



Michigan Technological University
Create the Future Digital Commons @ Michigan Tech

Dissertations, Master's Theses and Master's
Reports - Open

Dissertations, Master's Theses and Master's
Reports

2012

Computer Aided Design Algorithmic Technique for Very Large Scale Integrated Circuit Design and Smart Home Embedded System Design

Xiaodao Chen
Michigan Technological University

Follow this and additional works at: <https://digitalcommons.mtu.edu/etds>



Part of the [Computer Engineering Commons](#)

Copyright 2012 Xiaodao Chen

Recommended Citation

Chen, Xiaodao, "Computer Aided Design Algorithmic Technique for Very Large Scale Integrated Circuit Design and Smart Home Embedded System Design", Dissertation, Michigan Technological University, 2012.

<https://doi.org/10.37099/mtu.dc.etds/709>

Follow this and additional works at: <https://digitalcommons.mtu.edu/etds>



Part of the [Computer Engineering Commons](#)

COMPUTER AIDED DESIGN ALGORITHMIC TECHNIQUE FOR VERY LARGE
SCALE INTEGRATED CIRCUIT DESIGN AND SMART HOME EMBEDDED
SYSTEM DESIGN

By
Xiaodao Chen

A DISSERTATION

Submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

In Computer Engineering

MICHIGAN TECHNOLOGICAL UNIVERSITY

2012

J. ROBERT VAN PELT LIBRARY
MICHIGAN TECHNOLOGICAL UNIVERSITY
HOUGHTON, MICHIGAN

© 2012 Xiaodao Chen

This dissertation has been approved in partial fulfillment of the requirements for the Degree
of DOCTOR OF PHILOSOPHY in Computer Engineering

Department of Electrical and Computer Engineering

Dissertation Advisor: *Shiyan Hu*

Committee Member: *Zhuo Feng*

Committee Member: *Chee-Wooi Ten*

Committee Member: *Zhuo Li*

Department Chair: *Dan Fuhrmann*

APR 10 2014

For My Parents Qingjian Zou, Xiaoqin Chen and My wife Yuan

Liang

Contents

List of Figures	xv
List of Tables	xix
Preface	xxiii
Acknowledgments	xxvii
Abstract	xxix
1 Introduction	1
1.1 Motivation	1
1.2 Contributions	6

1.2.1 An Interconnect Reliability-Driven Routing Technique For Electromigration Failure Avoidance	7
1.2.2 A Transceiver Insertion Framework for On-Chip Optical Integration	8
1.2.3 Fault-Aware Energy Efficient Parallel Task Scheduling	10
1.2.4 Uncertainty-Aware Household Appliance Scheduling	11
2 An Interconnect Reliability-Driven Routing Technique For Electromigration Failure Avoidance¹	15
2.1 Introduction	15
2.2 Preliminaries	21

¹©2012 IEEE. Portions reprinted with permission, from **Xiaodao Chen**, Chen Liao Tongquan Wei and Shiyao Hu, “An Interconnect Reliability-Driven Routing Technique for Electromigration Failure Avoidance”, IEEE Transactions on Dependable and Secure Computing, vol. 9, pp.770-776, 2012.

2.3 Preliminaries	21
2.3.1 Interconnect Reliability	21
2.3.2 Problem Formulation	23
2.4 The Algorithm	29
2.4.1 Integer Linear Programming (ILP) Formulation	29
2.4.2 Handling Blockage	37
2.4.3 Iterative Rounding To Solve ILP	41
2.5 The Algorithm For Multiple Current Levels	43
2.6 Experimental Results	52
2.6.1 Optimality Study	53
2.6.2 Performance on Single Current Case	55
2.6.3 Performance on Multiple Current Case	59
2.7 Conclusion	61

3 A Transceiver Insertion Framework for On-Chip

Optical Integration² 63

3.1 Introduction 63

3.2 Preliminaries 73

3.2.1 Timing 76

3.2.2 Photon-Energy Loss 77

3.2.3 The Motivational Example 82

3.2.4 Problem Formulation 86

3.3 Photon-Energy Driven Transceiver-Aware Optical

Curved Tree Construction 90

3.3.1 Curved Tree Construction 92

3.3.2 Creating Candidate Transceiver Locations . . . 99

3.4 Dynamic Programming Based Transceiver Insertion 101

²©2010 IEEE. Portions reprinted with permission, from **Xiaodao Chen**, and Shiyan Hu, "A transceiver-aware routing framework for on-chip nanophotonic integration", in Proc. of IEEE Intl. Conf. on Intelligent Control and Information Processing (ICICIP), pp. 595-600, Aug. 2010.

3.4.1 Add Optical Wire	103
3.4.2 Add Transceiver	104
3.4.3 Merge Branch	106
3.4.4 Pruning	108
3.5 Experimental Results	110
3.6 Conclusion	119
 4 Fault-Aware Energy Efficient Parallel Task	
 Scheduling³	121
4.1 Introduction	121
4.2 System Architecture and Models	128
4.2.1 Energy Model	129
4.2.2 Model for Fault Estimation	132
4.2.3 Application and Recovery Model	134

³©2011 IEEE. Portions reprinted with permission, from Tongquan Wei, **Xiaodao Chen** and Shiyang Hu, “*Reliability-Driven Energy-Efficient Task Scheduling for Multiprocessor Real-Time Systems*”, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 30, pp. 1569-1573, 2011.

4.3 Integer Linear Program (ILP) Formulation	140
4.4 Reliability-Driven Parallel Task Scheduling for Multiprocessor Systems	149
4.4.1 β -Enabled Parallel Multiprocessor Task Scheduling	151
4.4.2 Generate Task Schedules Using a LP Solver and Sequential Rounding Technique	155
4.4.3 Derive Task Reliability Using the Monte Carlo Simulation and Latin-Hypercube Sampling	159
4.4.4 Complexity Analysis	167
4.5 Numerical Results	169
4.6 Conclusion	176

5 Smart Home Uncertainty-Aware Household

Appliance Scheduling⁴	179
5.1 Introduction	179
5.2 System Architecture and Models	195
5.2.1 Residential Customer Model	195
5.2.2 Pricing Model	201
5.2.3 Photovoltaic (PV) Model	203
5.3 The Proposed Stochastic Scheduling Algorithm	
For Household Appliances	205
5.3.1 Linear Programming based Deterministic	
Scheduling	206
5.3.2 Energy Adaptation Variable based Offline	
Stochastic Scheduling	214

⁴The material contained in this chapter has been accepted by IEEE Transactions on Smart Grid for publication.

5.3.2.1 β -Enabled Parallel Appliance	
Operation Scheduling	216
5.3.2.2 Derive the Trip Rate Using	
Monte Carlo Simulation and	
Latin-Hypercube Sampling	220
5.3.3 Adaptation of Offline Operation Schedules	
to Online Scheduling	224
5.4 Simulation Results and Discussions	230
5.5 Conclusions	240
6 Conclusions	245
References	249
Appendix	289
Appendix A Copyright Permission for Chapters 2-5 . .	289

List of Figures

2.1 Node current computation in LP formulation. . . .	33
2.2 A routing solution.	35
2.3 Two solutions with the same objective solution. . .	37
2.4 An example with blockage.	38
2.5 A routing solution with blockage.	40
2.6 Routes with different topologies by different current levels.	45
2.7 Illustration of different topologies in a multi-current problem.	47
2.8 An example with multi-current.	50

3.1 Illustration of optical integration.	69
3.2 Optical interconnect needs sufficient driving energy	70
3.3 The sources of photon-energy loss for optical integration.	78
3.4 Illustration of computing photon-energy loss of two connected lines.	81
3.5 A transceiver insertion example.	86
3.6 Illustration of a candidate transceiver location. . .	88
3.7 Optical Routing Procedures	96
(a) Photon-energy loss on curves v.s. on two line segments	96
(b) Chosen candidate curves	96
(c) All neighboring connections between two components	96

(d) The resulting curved tree	96
3.8 Tree construction avoiding coupler.	97
3.9 Illustration of branch merge.	109
3.10 Illustration of transceiver insertion.	111
4.1 Reliability-driven parallel task scheduling (RDPS)	151
4.2 Select the next candidates of β using the binary-search algorithm	153
4.3 Random sampling versus Latin Hypercube sampling	163
(a) Random sampling	163
(b) Latin Hypercube sampling	163
4.4 Adapt LH to discrete Poisson distribution	164
5.1 Home appliances electricity consumption in average	183
5.2 A smart home scenario	200

5.3 The Offline stochastic scheduling algorithm to iteratively derive the energy adaptation variable β and generate the monetary expense efficient operation schedule	216
5.4 The design flow of the household appliance energy consumption scheduling algorithm	217
5.5 Random sampling versus Latin Hypercube sampling	223
(a) Random sampling	223
(b) Latin Hypercube sampling	223
5.6 Compare the monetary expense of online and offline operation schedule	240

List of Tables

1.1 Technology trend for chips	2
2.1 Optimality study on 50 small testcases.	55
2.2 Comparison of previous work [1] and the new LP algorithm without blockage.	58
2.3 Comparison of previous work [1] and the new LP algorithm with blockage.	58
2.4 Result of multi-current without blockage.	60
2.5 Result of multi-current with blockage.	60
3.1 Delay (ps) of electrical and optical interconnects per unit length [2].	77

3.2 Delay (ps) of optical devices [2].	77
3.3 The power and area of each transceiver.	113
3.4 Comparison of tree constructions.	115
3.5 Comparison results after transceiver insertion. . .	118
3.6 Results on five selected nets which need to create the extra candidate transceiver locations	118
4.1 The average energy consumption (in mJ) of task sets with varying sizes and the average CPU time (in seconds) of the proposed RDPS algorithm ($0 <$ $\beta < 1$) for target reliability of 0.99.	166
4.2 Comparison of the proposed stochastic RDPS algorithm ($0 < \beta < 1$) with the best case ($\beta =$ 0) and worst case approach ($\beta = 1$) in energy consumption (in mJ). CPU time is in seconds. . .	166

5.1 Dynamic pricing model comparison [3].	185
5.2 Comparison of the deterministic LP-based approach and the traditional scheduling approach .	229
5.3 Comparison of the worst case, the best case, and the stochastic design in terms of monetary expense and scheduling time	229
5.4 Comparison of the proposed stochastic designing approach with the traditional scheduling approach	229



Preface

This dissertation presents my research work in pursuing the Ph.D. degree in Computer Engineering at Michigan Technological University. This dissertation includes previously published articles in Chapter 2, Chapter 3 and Chapter 4.

This dissertation includes articles which has been accepted in Chapter 5.

Chapter 2 contains an article previously published in IEEE Transactions on Dependable and Secure Computing. As the first author, I designed the algorithm flow for single current

level, multi-current level and blockage avoidance. The 80% coding part were completed by me. Under the guidance of my advisor Dr. Shiyan Hu (fourth author of this article) I completed the layer assignment part. With the collaboration with Ms. Chen Liao (second author of this article) and Dr. Tongquan Wei (third author of this article), I completed experimental part. This article was completed by me, Dr. Tongquan Wei, Ms. Chen Liao and Dr. Shiyan Hu.

Chapter 3 contains an article previously published in IEEE International Conference on Intelligent Control and Information Processing. As the first author, I designed the optical curved-aware optical routing tree. Then with the guidance of my advisor Dr. Shiyan Hu, I completed the transceiver insertion scheme. This article was completed by

me and Dr. Shiyan Hu.

Chapter 4 contains an article previously published in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. As the second author, I identified the linear programming formulation. With the collaboration with the first author Dr. Tongquan Wei, I completed stochastic programming part. Under the guidance of my advisor Dr. Shiyan Hu (third author of this article), I completed dependent task constraints in the linear programming formulation. This article was completed by Dr. Tongquan Wei, me and Dr. Shiyan Hu.

Chapter 5 contains an article has been accepted in IEEE Transactions on Smart Grid. As the second author, I identified the problem formulation. With the collaboration with the

second author Dr. Tongquan Wei, I completed the linear programming formulation. Under the guidance of my advisor Dr. Shiyan Hu (third author of this article), I completed the sampling method for task variations. I completed the online algorithm part. This article was completed by me, Dr. Tongquan Wei and Dr. Shiyan Hu.

Acknowledgments

I would like to thank my advisor Dr. Shiyan Hu for his most kind guidance and advice for my study and research. His guidance and support have carried me through my Phd career. I truly appreciate his encouragement, patient and enthusiasm to me.

I would also like to thank my dissertation members, Dr. Zhuo Feng, Dr. Chee-Wooi Ten and Dr. Zhuo Li for their very valuable assistance to my dissertation.

In addition, I would like to thank Dr. Zhuo Feng for spending time in discuss VLSI problems with me and made me learned a lot. I really appreciate Dr. Chee-Wooi Ten for discuss CAD

problems in power system. He gave me a great guidance and valuable comments. I would also like to thank the me great friends at Tech for their helps on my researches. They are Tongquan Wei, Sheng Hu, Chen Liao, Shuo Huang, Jia Wang, Xueqian Zhao, Yonghe Guo and Lin Liu.

Last but not least, I would like to thank my family for their encouragement and support!

Abstract

To tackle the challenges at circuit level and system level VLSI and embedded system design, this dissertation proposes various novel algorithms to explore the efficient solutions. At the circuit level, a new reliability-driven minimum cost Steiner routing and layer assignment scheme is proposed, and the first transceiver insertion algorithmic framework for the optical interconnect is proposed. At the system level, a reliability-driven scheduling scheme for real-time embedded systems, which optimizes multiprocessor system energy consumption under stochastic fault occurrences, is proposed. The embedded system design is also widely used in the smart home area for improving health, wellbeing

and quality of life. The proposed scheduling scheme for multiprocessor embedded systems is hence extended to handle the energy consumption scheduling issues for smart homes. The extended scheme can arrange the household appliances for operation to minimize monetary expense of a customer based on the time-varying pricing model.

Chapter 1

Introduction

1.1 Motivation

With fast technology scaling, VLSI technology has entered nanoscale regime. Increasing number of transistors results in large chip density [4]. The design shrinking feature makes the VLSI circuit complexity keep increasing. Therefore, Computer Aided Design (CAD) for VLSI design has to face

more challenges than ever before. One critical challenge is the design reliability. Research in [5] has shown that thermal profile has manifested increasingly stronger impacts to circuit reliability through threshold voltage, carrier mobility and saturation velocity. Electromigration is other major reliability concern which refers to the transport of material caused by ion movement in interconnects.

Table 1.1
Technology trend for chips

year	Transistor density (Million transistors / mm^2)	Gate Length (nm)	Chip Size (mm^2)	On-chip Clock (GHz)
2011	798	35	260	3.744
2012	1129	31	184	3.894
2013	1596	28	260	4.050
2014	2011	25	206	4.211
2015	2534	22	164	4.380

Besides the reliability, the interconnect delay also has become a critical issue. It is the dominating factor in VLSI circuit performance. A variety of issues such as increasing interconnect resistivity, low bandwidth and serious cross talks

have limited the usage of copper interconnect in the deep submicron node. As the emerging technology improving, on-chip optical waveguide emerges as a promising replacement material for copper interconnect. ITRS has projected and advocated the research on the computer aided design (CAD) for on-chip optical integration. However, there are very limited CAD techniques for On-chip Optical design.

Meanwhile, the transient faults number increases due to the increasing level of integration and reducing size of transistor features in addition to harsh operating environments [6]. The probability of fault occurrences is even higher in a multiprocessor system as a result of large number of components and increased design complexity. Since real-time applications demand system robustness, it is desirable that

in the presence of faults real-time tasks finish execution before their respective deadlines. Meanwhile the need for energy-efficient design is increasing for battery-powered real-time systems to reduce power density and enhance the system operational lifetime. Therefore the joint optimization of energy and fault-tolerance as two important design constraints for safety-critical multi-processor real-time systems is highly demanded.

The application of embedded controllers in smart home scenario has been strikingly leveraged as technology advances towards the deep submicron region. The kernel of the smart home applications lies in the demand side management that relies upon embedded optimization techniques. This dissertation proposes a novel demand side management scheme

for smart home applications. The new scheme arranges the household appliances for operation such that the monetary expense of a customer is minimized based on the time-varying pricing model.

To conclude, the optimization of VLSI circuit design and embedded system scheduling has become challenging with the advancement in technology, which in turn give rise to optimization issues at an even higher level of smart home designing. This dissertation explores effective and innovative CAD algorithmic solutions to design optimization issues in these areas.

1.2 Contributions

This dissertation addresses VLSI design, embedded system and smart home scheduling innovative algorithmic techniques. For reliability interconnect, a new reliability-driven minimum cost routing and layer assignment approach is proposed. For the optical interconnect, a new transceiver insertion algorithmic framework is proposed. For the embedded system scheduling, a reliability-driven task scheduling scheme is proposed. For smart home scheduling, a new energy efficient scheduling algorithm is proposed to arrange the household appliances.

1.2.1 An Interconnect Reliability-Driven Routing Technique For Electromigration Failure Avoidance

With fast technology scaling, a major design reliability concern arises from electromigration. In this project, the minimum cost reliability driven routing, which consists of Steiner tree construction and layer assignment, is considered. The problem is proven to be NP-hard and a highly effective iterative rounding based integer linear programming algorithm is proposed. In particular, the first algorithm to directly handle multiple currents is designed, which is critical in analog VLSI design due to the existence of various current levels. Further, the new algorithm is extended to handle blockage, which makes it ready for practical use. Our experiments on 450 nets

demonstrate that the new algorithm significantly outperforms the state-of-the-art work [1] with up to 14.7% wire reduction. In addition, the new algorithm can save 11.4% wires over a heuristic algorithm for handling multiple currents.

1.2.2 A Transceiver Insertion Framework for On-Chip Optical Integration

A variety of issues such as increasing interconnect resistivity, low bandwidth and serious cross talks have limited the usage of copper interconnect in the deep submicrometer node. On-chip optical waveguide emerges as a promising replacement material for copper interconnect. The deployment of on-chip optical integration in modern VLSI design certainly needs the advanced CAD tools. This work proposes the first

transceiver insertion algorithmic framework which is highly necessary to meet the energy demands in on-chip optical integration. The proposed algorithmic framework includes an efficient dynamic programming based transceiver insertion algorithm and a novel transceiver-aware tree construction algorithm for on-chip optical waveguide. Both algorithms are dedicated to optical integration which features the optimization of e.g., curvature and nanophotonic energy loss. Experimental results on 500 timing critical nets demonstrate the effectiveness and the efficiency of our techniques. The transceiver-aware on-chip optical tree construction algorithm can reduce the energy demand by $2.6\times$ compared to a natural minimum spanning tree heuristic. The transceiver insertion algorithm can always satisfy the energy demand while still minimizing

transceiver cost overhead. In fact, without transceiver insertion, 55.2% nets cannot meet the photon-energy demand. Further, our constructed optical trees can improve the timing by about $2\times$ compared to copper trees. These demonstrate the saliency of our algorithms.

1.2.3 Fault-Aware Energy Efficient Parallel Task Scheduling

This project presents a reliability-driven task scheduling scheme for real-time embedded systems that optimizes multi-core system energy consumption under stochastic fault occurrences. The task scheduling problem is formulated as an integer linear program where a novel fault adaptation variable is introduced to model the uncertainties of fault occurrences. The proposed scheme considers both the dynamic power and the

leakage power, is able to handle the scheduling of independent tasks and tasks with precedence constraints, and is capable of scheduling tasks with varying deadlines. Experimental results have demonstrated that the proposed reliability-driven parallel scheduling scheme achieves energy savings of more than 15% when compared to the approach of designing for the corner case of fault occurrences

1.2.4 Uncertainty-Aware Household Appliance Scheduling

This project presents a new demand side management technique, namely, a new energy efficient scheduling algorithm, is proposed to arrange the household appliances for operation such that the monetary expense of a customer is minimized based on the time-varying pricing model. The

proposed algorithm takes into account the uncertainties in household appliance operation time and intermittent renewable generation. Moreover, it considers the variable frequency drive and capacity-limited energy storage. Our technique first uses the linear programming to efficiently compute a deterministic scheduling solution without considering uncertainties. To handle the uncertainties in household appliance operation time and energy consumption, a stochastic scheduling technique, which involves an energy consumption adaptation variable β , is used to model the stochastic energy consumption patterns for various household appliances. To handle the intermittent behavior of the energy generated from the renewable resources, the offline static operation schedule is adapted to the runtime dynamic scheduling considering variations in renewable

energy. The simulation results demonstrate the effectiveness of our approach. Compared to a natural greedy algorithm which models typical household appliance operations in the traditional home scenario, the proposed deterministic linear programming based scheduling scheme achieves up to 53% monetary expense reduction. Compared to a worst case design where an appliance is assumed to consume the maximum amount of energy, the proposed stochastic design which considers the stochastic energy consumption patterns achieves up to 24% monetary expense reduction without violating the target trip rate of 0.5%. Furthermore, the proposed energy consumption scheduling algorithm can always generate the scheduling solution within 10 seconds, which is fast enough for household appliance applications.



Chapter 2

An Interconnect Reliability-Driven Routing Technique For Electromigration Failure Avoidance¹

2.1 Introduction

Design reliability becomes increasingly important in nanoscale circuit design. A major reliability concern is due to the

¹©2012 IEEE. Portions reprinted with permission, from **Xiaodao Chen**, Chen Liao Tongquan Wei and Shiyang Hu, “*An Interconnect Reliability-Driven Routing Technique for Electromigration Failure Avoidance*”, IEEE Transactions on Dependable and Secure Computing, vol. 9, pp.770-776, 2012.

electromigration which refers to the transport of material caused by ion movement in interconnects. The lifetime of an interconnect drastically depends on the current flowing through according to Black's Law [7]. Consequently, the problem aggravates when the wire becomes increasingly thinner. In addition, the current density-induced interconnect thermal issue becomes more severe with larger current density. To mitigate the electromigration effect and the induced thermal effect, it is in a great demand to reduce the current density along wires. Assigning wires to thick metals improves reliability due to their tolerance of large current density. However, over-stretching thick-metal assignment may make the design unroutable [8]. Thus, it is highly desirable to minimize the thick-metal usage, or total wire cost, subject to the constraint

on Mean Time To Failure (MTTF) which is a commonly-used metric of reliability.

There are some previous works addressing this reliability driven minimum cost Steiner routing and layer assignment problem [1, 9, 10, 11, 12]. However, they are mainly focused on applying the optimizations in the global routing stage. Since timing closure is already very difficult to achieve in advanced technology, it is not desirable to perform simultaneous timing and reliability driven global routing. This may necessarily put more burden to the already over-squeezed global routing stage. Thus, our application target is given to the detailed routing stage in a design flow for interconnect lifetime improvement. Note that previous works [1, 9, 13] perform wire sizing to approximate the layer assignment. In reality, only discrete

routing layers are presented and continuous wire sizing may not be practical. There are also works (e.g., [8]) on layer assignment for timing optimization, however, they do not consider the reliability issue.

In this project, a new reliability-driven minimum cost Steiner routing and layer assignment approach is proposed. In contrast to all the previous work that designs heuristics, our new algorithm is based on the integer linear programming formulation. This allows us to compute the optimal routing on small nets and greatly improve the solution quality on large nets when compared to the state-of-the-art schemes presented in [1]. A striking feature for the new algorithm is its capability to handle multiple currents. This is particularly important in analog VLSI design due to the existence of a large multitude

of current levels. In [1], a single current value is used to approximate the effects of multiple current values. This technique can not be extended to handle multiple current values due to its underlying limitation. In contrast, the proposed routing technique directly handles the multiple current values by a novel *unified multiple current routing* technique.

In this project, reliability-driven routing problem is formulated as simultaneous Steiner tree construction and layer assignment. The main contribution of this project is summarized as follows.

† The minimum cost reliability-driven Steiner routing and layer assignment problem is proven to be NP-hard.

† A highly effective iterative rounding-based integer linear programming algorithm is proposed, allowing us to

compute optimal solutions on small nets and well approximate the optimal solutions on large nets.

† Multiple currents are handled by the proposed unified multiple current routing technique, which is critical in analog VLSI design.

† The new algorithm is extended to handle blockage, which makes it ready for practical use.

† Our experiments on 450 testcases demonstrate that the new algorithm significantly outperforms the state-of-the-art work [1] with up to 14.7% wire reduction. In addition, the new algorithm can save 11.4% wire as compared to the heuristic algorithm for handling multiple currents.

The rest of the project is organized as follows. Section 3.2

formulates the reliability-driven routing problem. Section 2.4 describes the iterative rounding based integer linear programming approach for the problem. Section 2.5 describes the algorithm for handling multiple currents. Section 3.5 presents the experimental results with analysis. A summary of work is given in Section 4.6.

2.2 Preliminaries

2.3 Preliminaries

2.3.1 Interconnect Reliability

In this project, the interconnect reliability is measured by a commonly-used metric, namely, Mean Time To Failure

(MTTF). According to Black's Law [7], interconnect MTTF is given as

$$MTTF = \frac{A}{J^2} e^{\frac{E}{KT}}, \quad (2.1)$$

where A denotes the cross-section area of the interconnect, J denotes the current density, E denotes the activation energy, K denotes the Boltzmann constant, and T denotes the temperature. The above Black's Law clearly indicates the significance of current density in the interconnect reliability. In the design, one needs to suppress maximum current density to increase the interconnect lifetime.

2.3.2 Problem Formulation

The technique presented in this project is applied to net level. The input to the reliability-driven routing problem is a driver r and a set of n sinks, denoted by $S = \{s_1, s_2, \dots, s_n\}$, which belong to a net. We are to construct a Steiner tree (with layer assignment) over the driver and sinks such that the reliability constraint is satisfied. Due to layer assignment, we are also given a set of m routing layers as $L = \{l_1, l_2, \dots, l_m\}$. Given an edge e on a layer l , denote by $w(e, l), l(e, l), h(e, l), c(e, l), d(e, l)$ the width, length, height, cost, and delay of the edge, respectively.

Before defining wire cost and wire delay, note that the driver and sinks are associated with currents. Denote by I_a the value

of the current associated with node a . If the current is injected into the node, I_a is positive. Otherwise, I_a is negative. Denote by I_e the current flowing through an edge e . Note that there can be a set of currents associated with each sink/driver that forms a time-varying series (e.g., a sinusoidal waveform) in analog VLSI design. The driver/sink currents can be computed by either circuit simulation or manually attaching current values by designers [1].

Let us now link the MTTF metric to the layer assignment. We follow the formulation in [1]. As shown in [1], for a wire to have reasonable MTTF, the width of the wire layer $w(e, l)$ needs

to satisfy [1]

$$w(e, l) \cdot h(e, l) \geq \frac{I_e \cdot s}{J_{peak}}, \quad (2.2)$$

where I_e is the current flowing through the wire e , s is a guard-band factor to balance the MTTF and routing resource usage, J_{peak} is the maximum allowable peak current density determined by the technology, and $h(e, l)$ is the thickness of routing layer l . MTTF and s are highly correlated. Varying s , different tradeoff between routing resource usage and MTTF can be obtained. In practice, s is set to be within $[1.1, 1.2]$ as shown in [1].

Given a wire e at a layer l , the cost for a wire is generic.

In this project, to illustrate the effectiveness of the proposed technique, the wire cost is defined by wire volume, i.e., $c(e, l) = l(e, l) \cdot w(e, l) \cdot h(e, l)$. This is a commonly used metric in routing and layer assignment. Eqn. (2.2) says that $w(e, l) \cdot h(e, l) \geq \beta I_e$ where $\beta = \frac{s}{J_{peak}}$ is a constant with a given *MTTF* target. For a routing tree T , define the total wire cost as the sum of the costs for all edges in T . Given an s which corresponds to a constraint on *MTTF*, one aims to compute a Steiner routing and layer assignment solution such that the total wire cost is minimized.

In this project, since the reliability-driven routing technique needs to change the routing topology and the layer assignment, timing degradation may be introduced to the initial design. To evaluate the timing, the widely-used Elmore delay model

is adopted. That is, $d(e, l) = R_e \cdot (C_e/2 + C_l)$ where R_e, C_e, C_l are edge resistance, edge capacitance and load capacitance, respectively.

The problem is to meet an MTTF target (through fixing s) by Steiner tree construction and layer assignment with minimum total wire cost. It can be formulated as follows.

Reliability-Driven Minimum Cost Steiner Routing and

Layer Assignment: Given a driver and a set of sinks, each of which is associated with a current vector, a set of routing layers L , and cost of each wire on each layer, to compute a Steiner tree with layer assignment such that reliability constraint is satisfied and the total wire cost is minimized.

The problem is NP-hard. This can be shown by reducing from

minimum length Steiner tree problem which is known to be NP-hard [14]. Given an instance of a minimum Steiner tree problem, assign it as an instance for our problem where there is only one layer (with one size) and it can hold all possible currents. We are to compute a minimum volume Steiner tree, which is the same as a minimum length Steiner tree since wire width and wire height are constants in a single layer. Thus, this problem is NP-hard given that the minimum length Steiner tree problem is NP-hard. We have

Theorem 1: The reliability-driven minimum cost Steiner routing and layer assignment problem is NP-hard.

2.4 The Algorithm

2.4.1 Integer Linear Programming (ILP) Formulation

In the proposed algorithm for interconnect reliability-driven routing, both Steiner tree construction and layer assignment will be performed. Each wire/edge at a layer is associated with an MTTF constraint as computed by Eqn. (2.2). The problem is first formulated to an integer linear programming (ILP) problem and then solved by the iterative rounding technique. Previous works [15, 16] also focus on the ILP-based routing. However, none of them considers the reliability-driven current issues as handled in this project.

A Hannan grid is first constructed on the given set of the driver

and sinks. Each grid point is associated with a node. Some of them are driver and sinks. Our Steiner tree will only use Hannan edges which results in the commonly used Hannan routing. For simplicity, index each Hannan edge e by its two endpoints (a, b) which are Hannan grid points. In the ILP formulation, each Hannan edge is associated with a variable $I_{a,b}$ corresponding to the value of the current flowing through e .

By Kirchhoff's Law, for each node g , the sum of input currents is equal to the sum of output currents. That is, for a Steiner node s which is neither a driver nor a sink,

$$\sum I_{s-in} = \sum I_{s-out}, \forall \text{Steiner node } s. \quad (2.3)$$

For a driver d , denote by $I_{d-drive}$ the current flowing from the

driver, one has,

$$\sum I_{d-in} + I_{d-drive} = \sum I_{d-out}, \forall \text{ driver } d. \quad (2.4)$$

For a sink s , denote by I_{s-sink} the current flowing out of the sinks s , one has

$$\sum I_{g-in} = \sum I_{g-out} + I_{g-sink}, \forall \text{ sink } g. \quad (2.5)$$

According to the MTTF constraint computed by Eqn. (2.2), each wire e needs to satisfy $w(e, l) \cdot h(e, l) \geq \beta \cdot I(e)$. Denote by v_e the *unit volume* of a wire e at layer l which is computed as $w(e, l) \cdot h(e, l)$. Since the wire volume needs to be larger than

the current subject to the factor of β , we have

$$v_e \geq \beta I(e), \quad (2.6)$$

where β denotes $\frac{I_e \cdot s}{J_{peak}}$.

Note that $v_e = 0$ means that wire e is not used in the routing.

In practice, there are only limited number of layers (e.g., 8 in some advanced technology) and each wire layer has its own characteristics. Let m denote the number of layers. v_e cannot take arbitrary continuous values, and it needs to be chosen from the set $v_e \in \{V_1, V_2, \dots, V_m\}$, where V_i denotes the unit volume that corresponds to assigning wire e to layer i .

It is helpful to look at a simple example as shown in Figure 2.1.

If the node a is a Steiner node, then $I_{ba} + I_{ea} = I_{ad} + I_{ac}$. If the node a is a sink, then $I_{ba} + I_{ea} = I_{ad} + I_{ac} + I_{a-sink}$. If the node a is a drive, then $I_{ba} + I_{ea} + I_{d-drive} = I_{ad} + I_{ac}$. The MTTF constraint

βI_{ad} , and

$$v_{ad} \geq$$

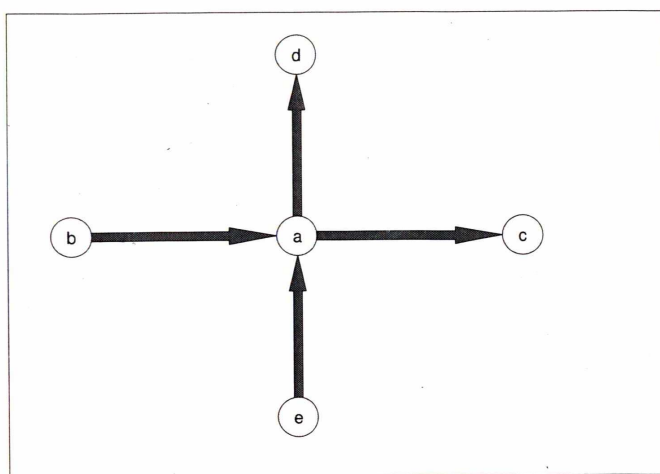


Figure 2.1: Node current computation in LP formulation.

The objective of the routing is to minimize the total cost of wires. The cost function is generic. To illustrate the effectiveness of the proposed technique, wire volume is used as

the cost which is a commonly-used metric. Precisely, wire cost is computed as $v_e \cdot l_e$ where l_e denotes the wire length (which is a constant if the edge is given) and v_e is the unit volume. The reliability-constrained minimum cost Steiner routing and layer assignment problem can be formulated to an integer linear program as follows.

$$\min \sum_e v_e \cdot l_e$$

s.t.

$$v_e \geq \beta l_e, \forall e$$

$$\sum I_{s-in} = \sum I_{s-out}, \forall \text{Steiner node } s \quad (2.7)$$

$$\sum I_{d-in} + I_{d-drive} = \sum I_{d-out}, \forall \text{driver } d$$

$$\sum I_{g-in} = \sum I_{g-out} + I_{g-sink}, \forall \text{sink } g$$

$$v_e \in \{V_1, V_2, \dots, V_m\}, \forall e.$$

After solving the ILP using standard LP solver, routing tree and layer assignment can be constructed by v_e . However, it is well known that the large-scale ILP cannot be efficiently solved due to the integer constraints. Thus, the classic iterative rounding technique is used in Section 2.4.3 to efficiently compute the integer solution.

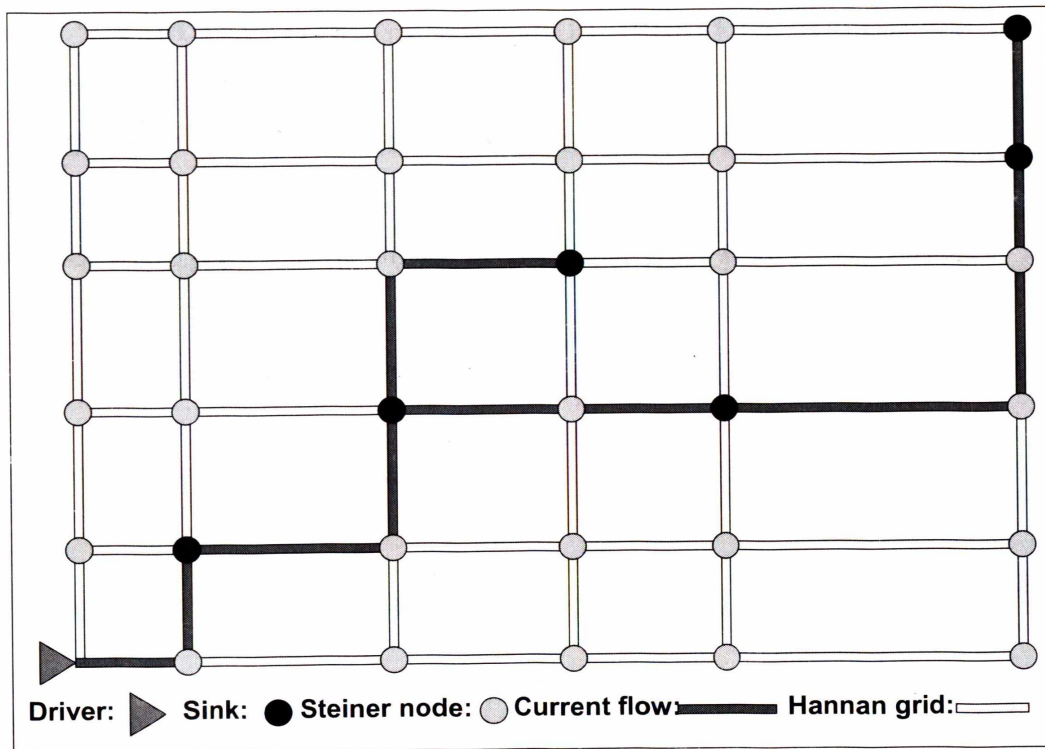


Figure 2.2: A routing solution.

Note that there can be multiple solutions for a net and the solution can be a non-tree. Consider the example shown in Figure 2.3. There are at least two solutions for this net (Figure 2.3(a) and Figure 2.3(b)). In the net, there are one driver, one sink and two Hanan nodes. For the simplicity, we assume that the width of layer 1 is 1, the width of layer 2 is 2 and lengths of all Hanan edges are equal to 1. Our LP program may produce either solution which has the same objective value 4. Therefore, for a net, there might be more than one solutions, and those solutions may be tree based (Figure 2.3(b)) or may be non-tree based (Figure 2.3(a)) because of the reconvergent path.

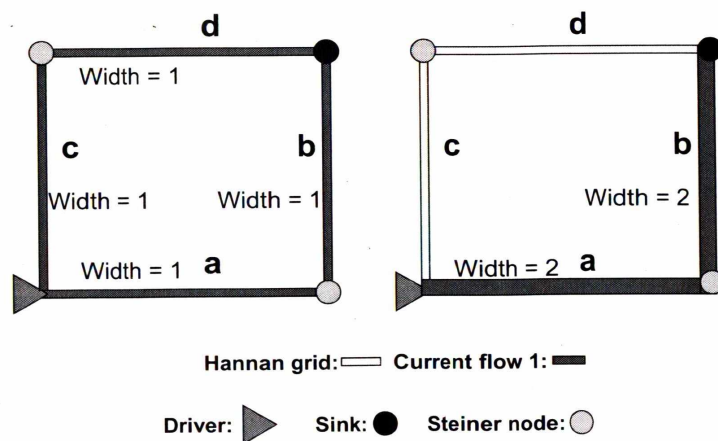


Figure 2.3: Two solutions with the same objective solution.

2.4.2 Handling Blockage

In VLSI design, there are often routing blockages (such as big macros) which need to be considered in constructing the route. Suppose that two blockages are introduced to Figure 2.2, which results in Figure 2.4. The blockage needs to be avoided in routing. To handle this, a weighting factor τ_e for each edge is introduced. Given an edge e , if it is blocked, set τ_e to

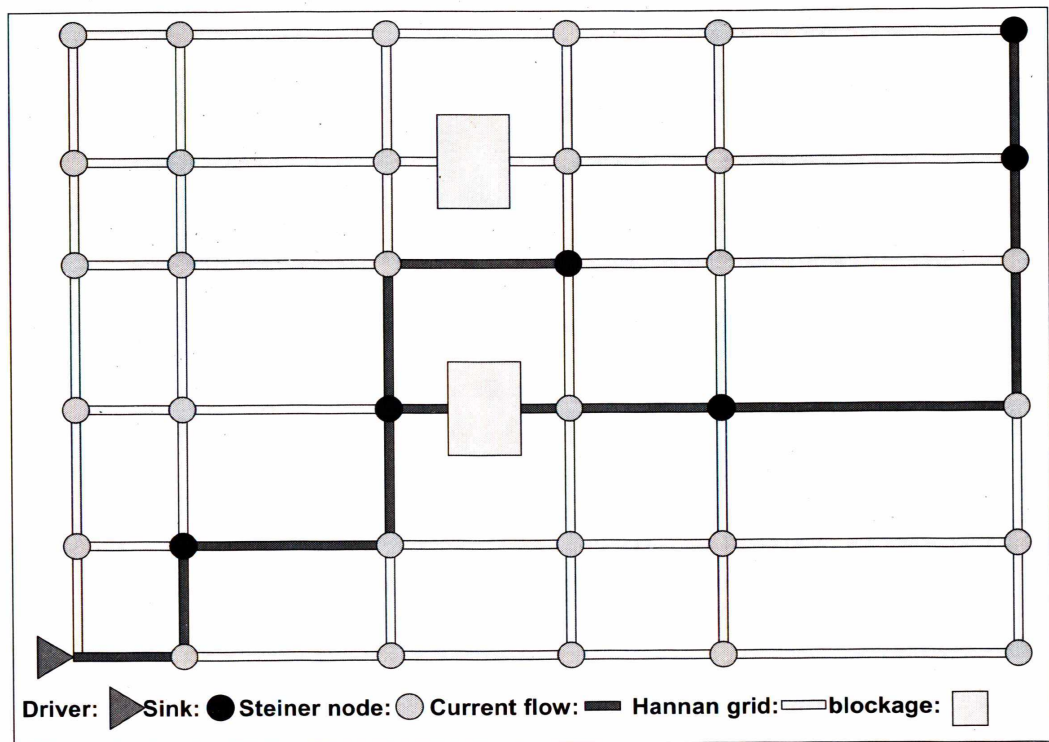


Figure 2.4: An example with blockage.

infinity. Otherwise, set τ_e to 1. In practice, a large number is used to serve as infinity. The objective function is changed to $\min \sum_e \tau_e \cdot v_e \cdot l_e$, where τ_e is the weighting factor on the edge e , while the remaining ILP remains the same as Eqn. (2.7). The new ILP is as follows.

$$\min \sum_e \tau_e \cdot v_e \cdot l_e$$

s.t.

$$v_e \geq \beta I_e, \forall e$$

$$\sum I_{s-in} = \sum I_{s-out}, \forall \text{Steiner node } s \quad (2.8)$$

$$\sum I_{d-in} + I_{d-drive} = \sum I_{d-out}, \forall \text{driver } d$$

$$\sum I_{g-in} = \sum I_{g-out} + I_{g-sink}, \forall \text{sink } g$$

$$v_e \in \{V_1, V_2, \dots, V_m\}, \forall e.$$

Figure 2.5 shows an example to illustrate a rerouted Steiner tree with blockage avoidance. Blocked edges are with large weighting factors. The part of the objective corresponding to edges (A,B) and (E,F) can be written as $\tau_{AB} \cdot v_{AB} \cdot l_{AB}$ and

$\tau_{EF} \cdot v_{EF} \cdot l_{EF}$, where $\tau_{AB} = 100k$ and $\tau_{EF} = 100k$. In contrast, the weighting factor of τ_e of unblocked edges is set to 1. For example, the part of the objective corresponding to edges (B,C) and (C,D) can be written as $\tau_{BC} \cdot v_{BC} \cdot l_{BC}$ and $\tau_{CD} \cdot v_{CD} \cdot l_{CD}$, where $\tau_{BC} = 1$ and $\tau_{CD} = 1$. The large τ_e prevents LP solver from choosing those edges in routing tree.

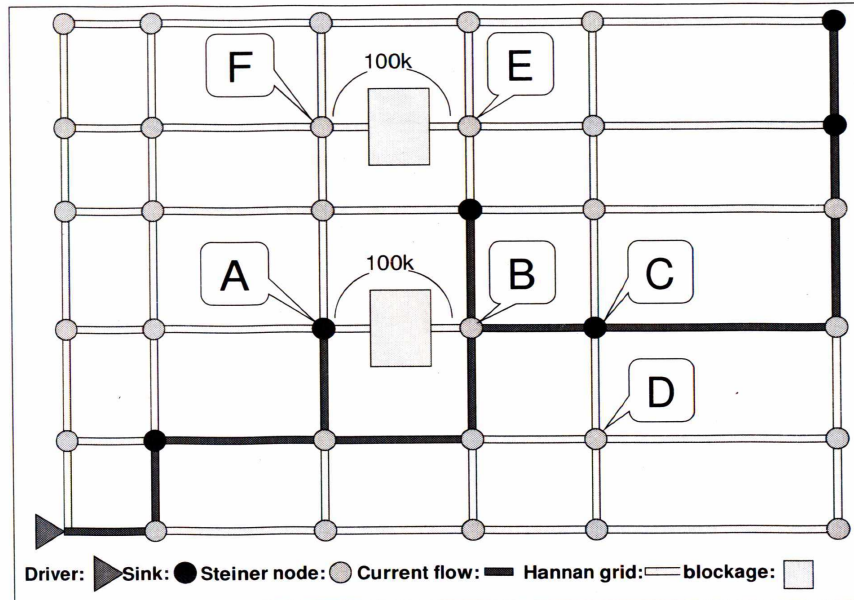


Figure 2.5: A routing solution with blockage.

2.4.3 Iterative Rounding To Solve ILP

Since there are many integer constraints in the linear programming formulation, the problem of the reliability-driven minimum cost Steiner routing and layer assignment cannot be efficiently solved on large nets. To handle this, commonly used iterative rounding technique is adopted. This technique has been used in a number of works in VLSI CAD such as [17].

As integer constraints $v_e \in \{V_1, V_2, \dots, V_m\}$ in Eqn. (2.7) are the most difficult constraints, they are relaxed as $0 \leq v_e \leq v_{max}$, where v_{max} denotes the largest value in $\{V_1, V_2, \dots, V_m\}$.

Consequently, the problem is translated into a *relaxed linear programming* problem which is well known to be much more efficient to solve. The iterative rounding is accomplished by

iteratively adding new constraints to the problem and solving the problem until all the variables have integer values.

First, the original relaxed linear program is solved, where the solution may contain fractional numbers. Define the feasible unit volumes as the unit volumes in $\{V_1, V_2, \dots, V_m\}$. All variables in the solution that are equal to the corresponding feasible unit volume v_t , where $v_t \in \{V_1, V_2, \dots, V_m\}$, will be first fixed to v_t . This is accomplished by adding new constraints such as $v_e = v_t$ to Eqn. (2.7) if v_e is equal to v_t in the solution. If there is no variable equal to any feasible unit volume, those close to v_t in the current solution will be chosen and fixed. Precisely, a *round up and fix* strategy will be used. That is, the variable with the rounding error (when being rounding up) smaller than δ , which is a user input, will be rounded up. If

there is no variable having rounding error smaller than δ , the variable with the smallest rounding error (when being rounded up) will be chosen, rounded up and fixed during the following iterations. In this way, the rounding procedure will converge and no oscillation may occur. The new linear program will be solved by LP again to obtain new solution. This process is iterated until all variables have feasible unit volumes. These strategies guarantee that in each iteration, there will be at least one new rounded variable which is fixed during following iterations and ensure that an integer solution will be obtained.

2.5 The Algorithm For Multiple Current Levels

The continuous-valued current in analog VLSI design is often quantized to discrete current values. Since the current flowing

through wires of an analog VLSI design varies at different instants, the discrete current values associated with sinks and the driver change with time. Let I_{t_i} denote the discrete current value at instant t_i for $1 \leq i \leq k$. The driver and sinks are associated with $n + 1$ discrete current values as $I_{t_i} = \{I_{1t_i}, I_{2t_i}, \dots, I_{nt_i}, I_{(n+1)t_i}\}$ where the driver is indexed as 1 and the sinks are indexed from 2 to $n + 1$. Since the topologies derived at each single current level are not guaranteed to be the same, directly applying the single current value technique described in Section 2.4 may not generate a feasible solution in the multi-current case. For example, the topology produced using ILP at the current level I_{t_1} may be different from the topology produced using ILP at the current level I_{t_2} . This can be clearly seen from Figure 2.6 by comparing blue route (for

I_{t_1}) and red route (for I_{t_2}).

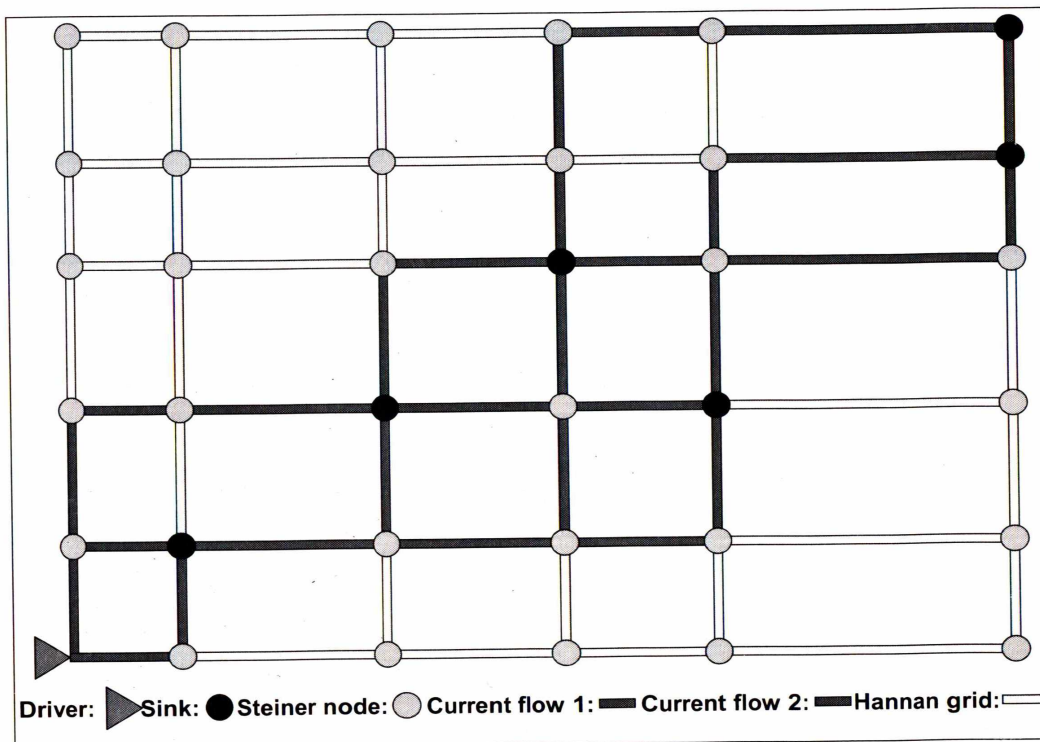


Figure 2.6: Routes with different topologies by different current levels.

An intuitive approach to the multi-current problem is to derive the routing topology at a certain current level and then adjust the layer assignment to satisfy the constraints corresponding to other current levels. However, this heuristic approach does not globally optimize the volume. Refer to Figure 2.7. Two sets

of current levels are given and they are shown as (I_{t_1}, I_{t_2}) where I_{t_1} corresponds to the first set of currents and I_{t_2} corresponds to the second set of currents. The computed topology of the first set of currents is shown in blue edges. In order to satisfy the constraint on the second set of currents, the heuristic algorithm needs to increase the layers on some blue edges as shown in Figure 2.7, e.g., from layer 1 to layer 2, and from layer 2 to layer 3. Blue edges will then satisfy the constraints of the both sets of currents and the total cost is 33. However, if we handle both sets of current levels simultaneously, the total cost is only 29, as shown in red edges. One can see that the topology and the layer assignment are significantly different between blue edges and red edges. This clearly demonstrates that the heuristic algorithm cannot compute the optimal routing

solution.

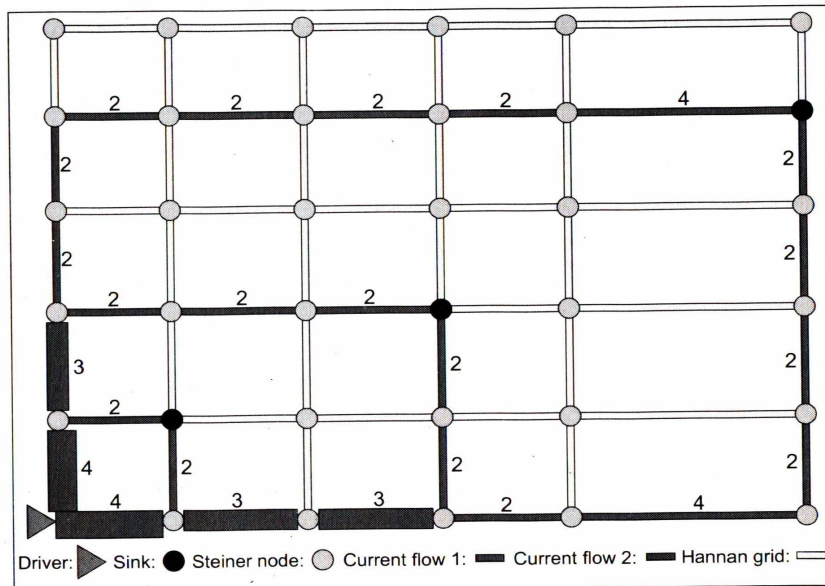


Figure 2.7: Illustration of different topologies in a multi-current problem.

In this project, a global optimization method is proposed for the multiple-current problem. In the new approach, all constraints at k current levels are incorporated in a unified fashion for any node including the driver, Steiner nodes and sinks. This is why our approach is called *unified multiple current routing*.

The example as shown in Figure 2.8 illustrates the proposed algorithm. Assume that each sink and driver have two current values. For the sink g with current values of -1 and -4, its constraints are given by

$$\begin{aligned}\sum I_{in\ t_1} &= \sum I_{out\ t_1} + 1, \\ \sum I_{in\ t_2} &= \sum I_{out\ t_2} + 4.\end{aligned}\tag{2.9}$$

The constraints of the the sterner node s are given by

$$\begin{aligned}\sum I_{in\ t_1} &= \sum I_{out\ t_1}, \\ \sum I_{in\ t_2} &= \sum I_{out\ t_2}.\end{aligned}\tag{2.10}$$

Similarly, the constraints on the driver d with current values 17 and 18 are given by

$$\begin{aligned}\sum I_{in\ t_1} + 17 &= \sum I_{out\ t_1}, \\ \sum I_{in\ t_2} + 18 &= \sum I_{out\ t_2}.\end{aligned}\tag{2.11}$$

The constraints on edges are the same as those in the single current value case, that is, $v_e \geq \beta I_{t_i}$.

The objective function of the multi-current optimization problem is the same as the single-current case. The problem

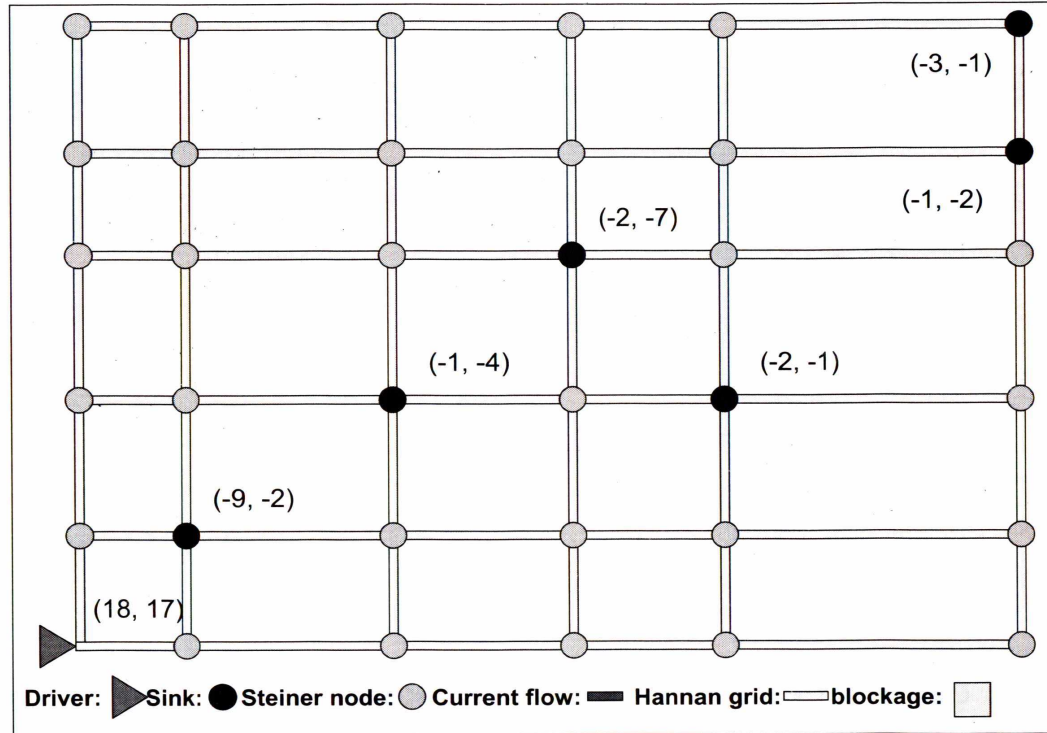


Figure 2.8: An example with multi-current.

formulation based on integer linear program is as follows.

$$\min \sum_e v_e \cdot l_e$$

s.t.

(2.12)

$$v_e \geq \beta I_e, \forall e$$

$$\sum_{I \text{ flows in } s} I_{t_i} = \sum_{I \text{ flows out of } s} I_{t_i}, \forall t_i, \text{ Steiner node } s$$

$$\sum_{I \text{ flows in } d} I_{t_i} + I_{driver \ t_i} = \sum_{I \text{ flows out of } g} I_{t_i}, \forall t_i, \text{ driver } d$$

$$\sum_{I \text{ flows in } g} I_{t_i} = \sum_{I \text{ flows out of } g} I_{t_i} + I_{sink \ t_i}, \forall t_i, \text{ sink } g$$

$$v_e \in \{V_1, V_2, \dots, V_m\}, \forall e.$$

As indicated by our experimental results, the new approach achieves area savings of up to 11.4% compared to the heuristic approach described in the beginning of this section. In addition, our technique are extended to deal with blockage, as is described in Section 2.4.2.

2.6 Experimental Results

The proposed algorithm for the reliability-constrained Steiner routing problem is implemented in C, and tested on a Pentium IV machine with 2.4GHz CPU and 8GB memory. Experiments are performed on a set of 450 nets at various scales. Due to the lack of the access to industrial analog nets, we randomly generated these 450 nets. The generated nets have varying numbers of sinks, current values, and coordinates. The number of sinks is in the range from 1 to 30. The number of Hanan edges varies from 4 to 3000, and the current values vary from 1 mA to 40mA. Since blockage is crucial in practical applications, we also investigate the algorithm performance with blockage. For those nets with blockage, we randomly

generate 1 to 20 blockages on each net. In this project, to illustrate the effectiveness of the proposed technique, the wire cost is defined by wire volume. However, our approach can also be extended to handle other cost metrics. The proposed algorithm in Section 2.4 is denoted by NEW and the proposed algorithm for handling multiple current levels in Section 2.5 is denoted by NEW w/ multi-current. Note that the shown Cost and Delay values are averaged over 450 nets.

2.6.1 Optimality Study

An optimality study is performed to validate the proposed algorithm. NEW takes randomly generated small nets as the input without considering blockage. Various experiments have been performed and 50 nets with 3-5 pins are chosen. In

Table 2.1 AVG. of 50 nets refers to averaged results on 50 nets. Results on 5 representative nets are also shown. CPU refers to the runtime in seconds. Refer to Table 2.1 for the results. The following observations are made.

† Due to the small size of the net, we can afford to solve ILP exactly and the optimal average cost is 41.

† NEW also works well on small nets. One can see the average cost by NEW is 42, which implies that NEW can almost optimally solves the problem on our small nets. No significant discrepancy in the runtime has been found. This is primarily due to that both algorithm can run very fast on small nets. However, for large nets, ILP becomes computationally prohibitive while NEW can still

solve them efficiently, which will be demonstrated in the next two sections.

Table 2.1
Optimality study on 50 small testcases.

Net	Algorithm	Cost	CPU(s)
Net1	Integer Linear Programming	35	0.002
	NEW (LP+Iterative Rounding)	36	0.002
Net2	Integer Linear Programming	40	0.002
	NEW (LP+Iterative Rounding)	41	0.002
Net3	Integer Linear Programming	43	0.002
	NEW (LP+Iterative Rounding)	43	0.002
Net4	Integer Linear Programming	29	0.002
	NEW (LP+Iterative Rounding)	30	0.002
Net5	Integer Linear Programming	47	0.002
	NEW (LP+Iterative Rounding)	48	0.002
AVG. of 50 nets	Integer Linear Programming	41	0.002
	NEW (LP+Iterative Rounding)	42	0.002

2.6.2 Performance on Single Current Case

We would compare our NEW algorithm to ILP, as is in the above section, on large nets. However, ILP becomes computationally prohibitive for handling most of the nets. Thus, we choose to compare NEW with the state-of-art

algorithm in [1]. The experimental results without blockage are shown in Table 2.2 and the results with blockage are shown in Table 2.3. In these two tables CPU refers to the runtime in seconds. Cost and Delay refer to average value per net. Ratio is computed as 1 minus the ratio of the cost or delay of NEW to that of [1]. The following observations are made.

† Without considering blockage, NEW exhibits better performance in the cost compared to [1] (refer to Table 2.2), especially for the large nets. For example, for the net with 21-30 sinks, NEW achieves $1 - 12822/14680 = 12.7\%$ reduction in cost compared to [1].

† With blockage, NEW also outperforms [1] (refer to Table 2.3) in cost. For example, for large nets, NEW achieves up to $1 - 12900/15128 = 14.7\%$ reduction in

cost.

† NEW exhibits better performance in the delay for most nets compared to [1]. For large nets, NEW achieves up to 44.9% saving in delay. Although the results of NEW for the small nets are not as good as those of [1], there are only slight differences. For instance, the difference is only 1.4% for the nets with blockage. NEW significantly outperforms [1] in delay when the size of the net increases.

† Although the optimal ILP solutions cannot be computed, the lower bound of the optimal solutions can be obtained by solving the relaxed LP. NEW can compute solutions much closer to the optima in both the cases with and without blockage compared to [1].

† There exists a tradeoff between the solution quality and

runtime. It is true that [1] saves some runtime over NEW. However, its solution quality is much worse. As discussed above, the reduction in the cost by NEW is up to 14.7% and the reduction in delay by NEW is up to 44.9%. The significant improvement in solution quality by NEW clearly outweighs the runtime slowdown.

Table 2.2
Comparison of previous work [1] and the new LP algorithm without blockage.

Sink #	Lower Bound			Previous Work [1]			NEW			Ratio	
	Cost	Delay	CPU(s)	Cost	Delay	CPU(s)	Cost	Delay	CPU(s)	Cost	Delay
1-10	2626	29.6	0.01	2938	34.6	0.0002	2826	37.5	32.2	3.8%	-8.4%
11-20	7442	30.0	0.08	8719	64.6	0.001	7784	38.0	613.3	10.7%	41.2%
21-30	12169	26.3	0.21	14680	65.4	0.001	12822	36.2	2314.1	12.7%	44.6%

Table 2.3
Comparison of previous work [1] and the new LP algorithm with blockage.

Sink #	Lower Bound			Previous Work [1]			NEW			Ratio	
	Cost	Delay	CPU(s)	Cost	Delay	CPU(s)	Cost	Delay	CPU(s)	Cost	Delay
1-10	2634	29.8	0.02	3103	36.4	0.0003	2890	36.9	0.221	6.9%	-1.4%
11-20	7444	30.0	0.09	9145	57.0	0.001	7855	38.2	4.491	14.1%	33.0%
21-30	12315	26.7	0.21	15128	68.0	0.001	12900	37.5	15.897	14.7%	44.9%

2.6.3 Performance on Multiple Current Case

The proposed algorithm NEW w/ multi-current is then validated for the multi-current case, which is important in analog VLSI design. To the best of our knowledge, this is the first work which directly handles the multiple current levels. NEW w/ multi-current is compared to the heuristic algorithm as described in the beginning of Section 2.5. The results are summarized in Table 2.4 and Table 2.5. Note that in these two tables, CPU refers to the runtime in seconds. We make the following observations.

† As shown in Table 2.4, without blockage, NEW significantly outperforms the heuristic algorithm. For example, for the net with 11-20 sinks, NEW achieves the

cost reduction by $1 - 6507/7320 = 11.1\%$ compared to the heuristic algorithm.

† With blockage, NEW is also much better than the heuristic algorithm, especially in large nets. For example, for the nets with 11-20 sinks, NEW achieves the cost reduction of 11.4%.

Table 2.4
Result of multi-current without blockage.

sink number	Heuristic algorithm		NEW w/ multi-current		Ratio
	Cost	CPU(s)	Cost	CPU(s)	Cost
1-10	2432	21	2303	43	5.3%
11-20	7320	62	6507	84	11.1%
21-30	12132	232	10820	271	10.8%

Table 2.5
Result of multi-current with blockage.

Sink number	Heuristic algorithm		NEW w/ multi-current		Ratio
	Cost	CPU(s)	Cost	CPU(s)	Cost
1-10	2485	23	2379	46	4.3%
11-20	7391	63	6548	85	11.4%
21-30	12697	246	11241	288	11.5%

2.7 Conclusion

With fast technology scaling, a major design reliability concern arises from electromigration. In this project, the minimum cost reliability driven routing, which consists of Steiner tree construction and layer assignment, is considered. The problem is proven to be NP-hard and a highly effective iterative rounding based integer linear programming algorithm is proposed. In particular, the first algorithm to directly handle multiple currents is designed, which is critical in analog VLSI design due to the existence of various current levels. Further, the new algorithm is extended to handle blockage, which makes it ready for practical use. Our experiments on 450 nets demonstrate that the new algorithm significantly outperforms

the state-of-the-art work [1] with up to 14.7% wire reduction.

In addition, the new algorithm can save 11.4% wires over a heuristic algorithm for handling multiple currents.

Chapter 3

A Transceiver Insertion Framework for On-Chip Optical Integration¹

3.1 Introduction

With fast technology scaling, interconnect delay has become the dominating factor in circuit performance. A variety of issues such as increasing interconnect resistivity, low

¹©2010 IEEE. Portions reprinted with permission, from Xiaodao Chen, and Shiyang Hu, “A transceiver-aware routing framework for on-chip nanophotonic integration”, in Proc. of IEEE Intl. Conf. on Intelligent Control and Information Processing (ICICIP), pp. 595-600, Aug. 2010.

bandwidth and serious cross talks have limited the usage of copper interconnect in the deep submicron node. On-chip optical waveguide emerges as a promising replacement material for copper interconnect due to various reasons such as the absence of RLC impedance in optical waveguide [18, 19]. A number of research efforts have been devoted to on-chip optical interconnect design. In the area of optical interconnect based Network-On-Chip (NoC) design, a low power optical transceiver has been proposed in [20], an optical 4×4 router is designed in [21] and a hybrid photonic NoC architecture is proposed in [22]. In fact, research in the on-chip optical communication has been conducted for quite a few years [23, 24, 25, 26, 27, 28, 29]. On the other hand, on-chip optical devices and waveguides have been successfully built on silicon

as indicated in [30],[31] and [28]. For example, on-chip optical modulator is implemented in [32], on-chip photodetector is implemented in [33], light source is implemented in [34], optical waveguides is implemented in [35] and [32], the lasing behavior between resonators and waveguide is studied in [36], the measurement on the bending effect is provided in [37], and the nickel based metal mold is fabricated in [38]. These demonstrate the potential of the practical application of on-chip optical integration.

ITRS has projected and advocated the research on the computer aided design (CAD) for on-chip optical integration. Multiple research works exist in this area. An architecture of integration of optical interconnects and CMOS devices is described in [39]. Such an architecture is especially useful for routing

timing critical nets using optical interconnects which are in an extra optical routing layer. An important issue in optical waveguide routing is how to connect an optical interconnect with a logic gate. For this, one needs to first convert electrical signal to optical signal (light) for transmission and later convert the optical signal back to electrical signal (refer to Figure ?? for this process). This methodology needs the usage of transmitter and receiver which themselves are the sources of area and power. Consequently, it is desired to limit their usage. On the other hand, existing works on on-chip optical interconnect routing include [40, 30]. [40] proposes the first on-chip optical routing algorithm but it only considers routing two-pin nets. [30] proposes an optical routing framework for general trees. However, their technique is based on integer

linear programming which would lead to large computational overhead in large circuits. More importantly, their technique does not consider creating *candidate locations for transceiver insertion* and the resulting routing tree may not satisfy the energy demands, where a transceiver denotes the combination of a receiver and a transmitter.

Refer to Figure 3.2(a) for a simple circuit demonstrating this problem, which also motivates this work. When traversing along an optical waveguide, there is significant photon-energy loss. Therefore, the driver (in the transmitter) which drives the optical interconnect must provide sufficient *driving energy* to compensate for the loss such that the optical signal can reach the destination. In Figure 3.2(a), there is just one transmitter at the source node of the circuit and thus there is only one

driver. Suppose that this driver can provide up to 8-unit photon-energy. Subsequently, the optical signal cannot reliably reach the destination since there is 10-unit photon-energy loss along the optical wire. In contrast, if we consider inserting a transmitter (together with a receiver) in the middle of the optical interconnect (refer to Figure 3.2(b)), there will be an additional driver which can also provide 8-unit photon-energy. After this, one clearly sees that photo-energy loss along each of the two optical wire segments can be compensated by the corresponding driver. Without creating extra candidate location for transceiver insertion, this is not achievable. The details of the parameter setting for such examples are discussed in Section 3.2.3.

The preliminary work of this project proposes a

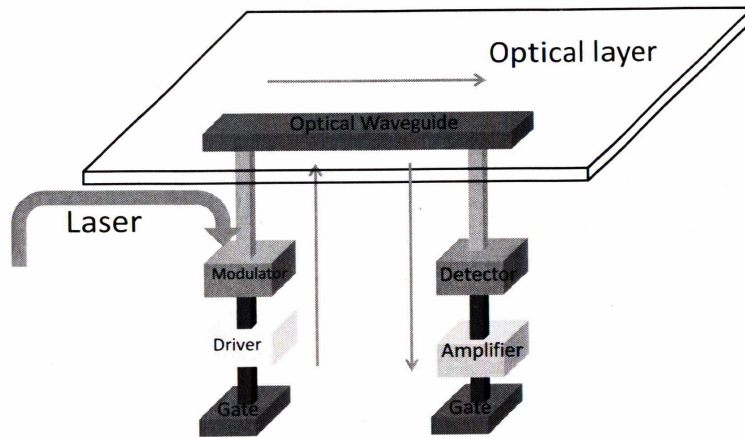


Figure 3.1: Illustration of optical integration.

transceiver-aware optical curved tree construction [41]. However, no transceiver is inserted in the optical curved tree in [41]. That is, [41] considers only routing while this work considers both routing and transceiver insertion. In this work, we significantly extend [41] through designing a new dynamic programming based transceiver insertion algorithm to insert transceivers into an optical curved tree. The new algorithm leads to much better energy and timing while still running fast. Note that both of the optical curved tree

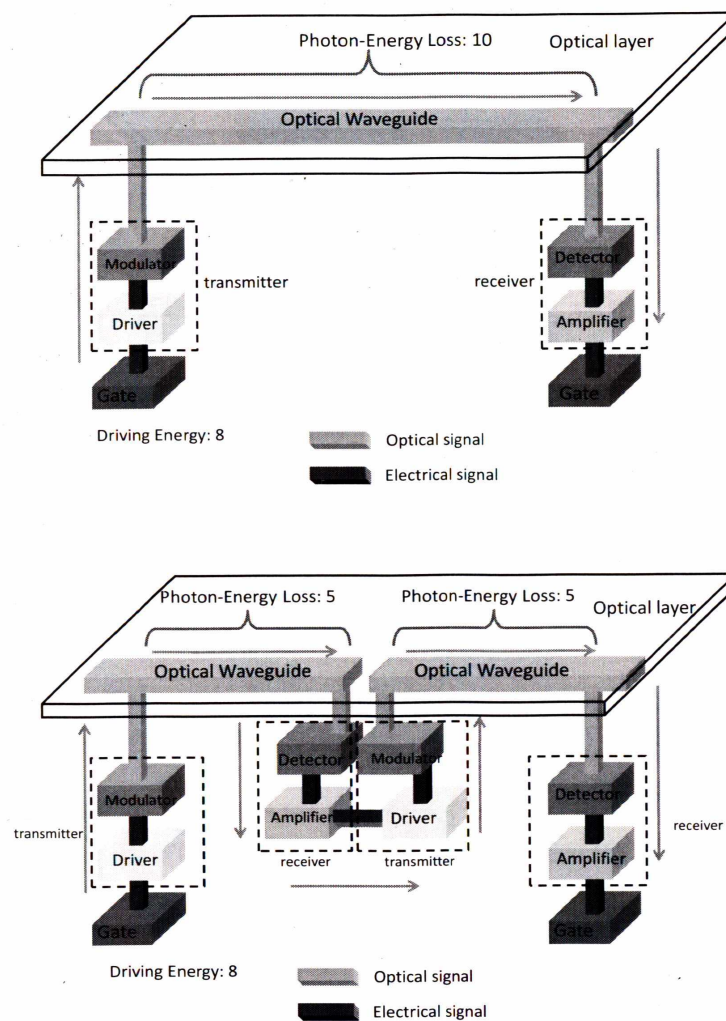


Figure 3.2: Optical interconnect needs sufficient driving energy

construction and the transceiver insertion would be important in the emerging physical syndissertation of on-chip optical integration. Comparing to [41], the dynamic programming

based transceiver insertion algorithm (Section 3.4) and the related experiments in Section 3.5 are new. The main contribution of the project is summarized as follows.

† The first transceiver insertion algorithmic framework is proposed which is highly necessary to meet the energy demands in on-chip optical integration.

† The proposed algorithmic framework includes an efficient dynamic programming based transceiver insertion algorithm and a novel transceiver-aware routing tree construction algorithm for on-chip optical waveguide. Both algorithms are dedicated to optical integration which features the optimization of e.g., curvature and nanophotonic energy loss.

† Experimental results on 500 timing critical nets demonstrate the effectiveness and the efficiency of our techniques. The transceiver-aware on-chip optical tree construction algorithm can reduce the energy demand by $1.9\times$ compared to a natural minimum spanning tree heuristic. The transceiver insertion algorithm can always satisfy the energy demand while still minimizing transceiver cost overhead. In fact, without transceiver insertion, 55.2% nets cannot meet the photon-energy demand. Further, our constructed optical trees can improve the timing by about $2\times$ compared to copper trees. These demonstrate the saliency of our algorithms.

The rest of the project is organized as follows: Section 3.2 describes the transceiver insertion. Section 3.3 presents

the transceiver-aware optical tree construction algorithm. Section 3.4 describes the dynamic programming based transceiver insertion algorithm for on-chip optical integration. Section 3.5 presents the experimental results with analysis. A summary of work is given in Section 4.6.

3.2 Preliminaries

In order to route signals originating from a logic gate through optical waveguides, one needs to first convert electrical signal to optical signal (light) for transmission and later convert optical signal back to electrical signal. This process can be illustrated using Figure ???. In Figure ??, at the transmitter side which consists of a driver and an electro-optical modulator, the electrical signal from the driver first modulates the light source

from the laser in the optical modulator, and then the modulated optical signal is sent to the optical waveguide. At the receiver side which consists of an amplifier and a photon detector, the photon-detector detects the photons from the optical waveguide and converts it to electrical signal [30].

In Figure ??, the upstream gate is called the driving gate, and the downstream gate is called the sink gate. In optical circuits, each of the driving gates needs to be connected to a transmitter, and each of the sink gates needs to be connected to a receiver. On the other hand, one may intentionally insert a transceiver at a candidate transceiver location since this may be necessary to meet energy demand (through providing more driving energy), which will be discussed in Section 3.2.2.

Refer to Figure 3.2(b) for a transceiver which is a pair of transmitter and receiver. In Figure 3.2(b) (also Figure ??), the green bar denotes optical signal and the black bar denotes electrical signal. One can see that the input of a transmitter is an electrical signal, while that of a receiver is an optical signal. The output of a transmitter is an optical signal and that of a receiver is an electrical signal. While only the driver in the transmitter provides driving photon-energy (to compensate for energy-loss along optical wire), we need to use a pair of transmitter and receiver (i.e., a transceiver) such that both input and output signals are optical.

Timing and photon-energy loss are two main concerns in on-chip optical integration and we will discuss them in turn.

3.2.1 Timing

The sources of delay in optical integration consist of two parts, i.e., the optical interconnect and the optical devices including transmitter and receiver. For the interconnect delay, Table 3.1 from [2] shows the parameters for different technology nodes. For the device delay, Table 3.2 from [2] shows the delay of each optical component (driver, modulator, waveguide, detector and amplifier). For a certain optical path, the delay can be calculated by summing up the delay of optical components. For the copper interconnects, we construct the Steiner tree as the copper routing and the parameters from [2] are used to calculate the delay.

Table 3.1

Delay (ps) of electrical and optical interconnects per unit length [2].

Technology Node	90nm	65nm	45nm	32nm	22nm
Electrical	311.9	313.2	291.3	312.0	317.8
Optical	238.9	173.3	145.4	127.7	114.9

Table 3.2

Delay (ps) of optical devices [2].

Technology Node	90nm	65nm	45nm	32nm	22nm
Driver	37.3	26.5	16.6	10.3	5.2
Modulator	40.0	40.0	40.0	40.0	40.0
Waveguide	49.3	49.3	49.3	49.3	49.3
Detector	2.5	1.1	0.6	0.5	0.4
Amplifier	34.0	13.5	8.7	5.7	3.4

3.2.2 Photon-Energy Loss

In on-chip optical integration, a critical issue is photon-energy loss, which is an analogue to power consumption in electrical interconnect. Similar to minimizing power in the conventional circuit optimization, minimizing photon-energy loss in optical integration while satisfying the timing constraint needs to be considered.

Given an optical wire, the amount of photon-energy loss depends on the shape of the wire. Different from the electrical interconnect routing typically in the form of rectilinear Steiner trees, optical interconnect routing is in the form of *curves*, possibly with the usage of couplers. The energy loss in all of these components needs to be considered.

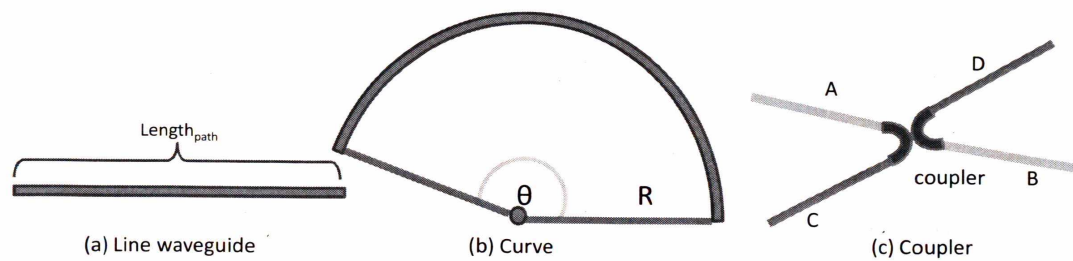


Figure 3.3: The sources of photon-energy loss for optical integration.

As shown in Figure 3.3, in an optical interconnect, there are

three types of photon-energy loss. The first one is the straight line waveguide loss, denoted by L_p . It is proportional to the length of the wire. The second one is the bending loss, denoted by L_b . It is proportional to the degree of the optical interconnect arc angle θ , and inversely proportional to the radius of the interconnect τ . The third one is the coupling loss, denoted by L_c . It is proportional to the number of couplers [30]. They can be computed as follows.

$$L_p = a \cdot l_{path}, \quad (3.1)$$

$$L_b = b \cdot g(\theta) \cdot h(\tau), \quad (3.2)$$

$$L_c = c \cdot \text{Couplers}, \quad (3.3)$$

where a , b and c are technology-dependant constants, and $g(\cdot)$ and $h(\cdot)$ are technology-dependant functions. Given a set of gates, different optical tree topology can lead to very different photon-energy loss. We are interested in computing a *curved tree* with minimum photon-energy loss. In a curved tree, an edge could be a curve (or a degenerate curve, i.e., a line). On-chip optical interconnect is an emerging technology, which makes it difficult to obtain accurate photon-energy loss estimation. The above photon-energy loss estimation equations used in this project are originally proposed in [30]. As claimed in [30], one can adapt the involved parameters to handle

different optical technologies.

Note that the above equations can be applied to compute photon-energy loss for all topologies. For example, given two connected line segments in Figure 3.4, their photon-energy loss can be obtained by summing up the photon-energy loss along the line segments, computed by Eqn. (3.1), and that of the turning point (which can be considered as a small curve with the sharp

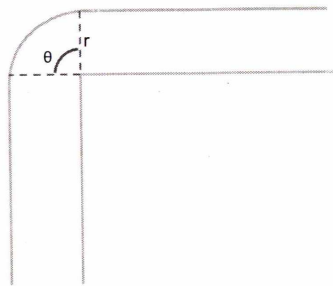


Figure 3.4: Illustration of computing photon-energy loss of two connected lines.

3.2.3 The Motivational Example

The photon-energy loss can be compensated through inserting transceivers. Refer to Figure 3.5 for a simple example which illustrates this procedure. We first discuss the parameter setting in the example.

† According to [36], the optical power provided by the laser is 5mW.

† According to [19], to reliably detect 1 bit data transmission, the minimum optical power at detector side has to be greater than 10 fJ ($10^{-14}J$) with $0.7 \mu m$ wavelength. We call this value (10 fJ) *threshold of optical detection*. If the optical power at detector side is less than the threshold

of optical detection, the noise will dominate the optical signal there, which means that the optical signal cannot be correctly detected. Suppose that one wants to transmit the data at 0.4 T (0.4×10^{12}) bit rate. According to the above, the power demand at the optical detector side in 1 second is $(0.4T \times 10^{12}) \times (10^{-14}J) = 4 \times 10^{-3}J$ which is equivalent to 4mW power. This is the threshold for correctly detecting these data.

† According to [35], the optical energy loss is 6db/cm if we use the waveguide that has 20nm slabs of gold and a 12 μm mode diameter, which is a typical setting.

† According to [39], the optical energy loss of the Mach-Zehnder Optical Modulator is 7db.

In Figure 3.5, there is a 1 cm waveguide, and thus the optical power loss is 6 db. In Figure 3.5(a), there is only one optical transceiver which is located at the left end of the net. The optical signal cannot be correctly detected at the detector side (right end) due to the power loss. Precisely, the total power loss is the summation of the optical power loss due to the modulator and that due to the waveguide. According to the above parameter setting, it is $7 + 6 = 13$ db, i.e., about 95% degradation. Thus, the power at detector side is only $5 \times (1 - 95\%) = 0.25$ mW which is lower than the threshold of detecting our data (0.4 mW as mentioned above). Thus, we need to insert an extra transceiver in the middle of the two-pin net, resulting in Figure 3.5(b). After the transceiver insertion, the detector at the middle of the waveguide can successfully detect

the optical signal. This is due to that the power loss between the left end to the middle point is $7 + 3 = 10$ db which is roughly 90% degradation. Thus, the power at the middle point is $5 \times (1 - 90\%) = 0.5$ mW which is greater than the threshold 0.4 mW. Due to the similar reason, the optical signal can be correctly detected at the right end of the waveguide. Therefore, in the above example inserting a transceiver in the middle of the wire can make the optical signal correctly detected.

While the technique proposed in [30] is quite effective in computing low-energy-loss routing trees for on-chip optical integration for many scenarios, there could be some scenarios where the optical signals cannot be correctly detected similar to the above example. Thus, this work proposes the transceiver insertion technique for those scenarios. In other words, our

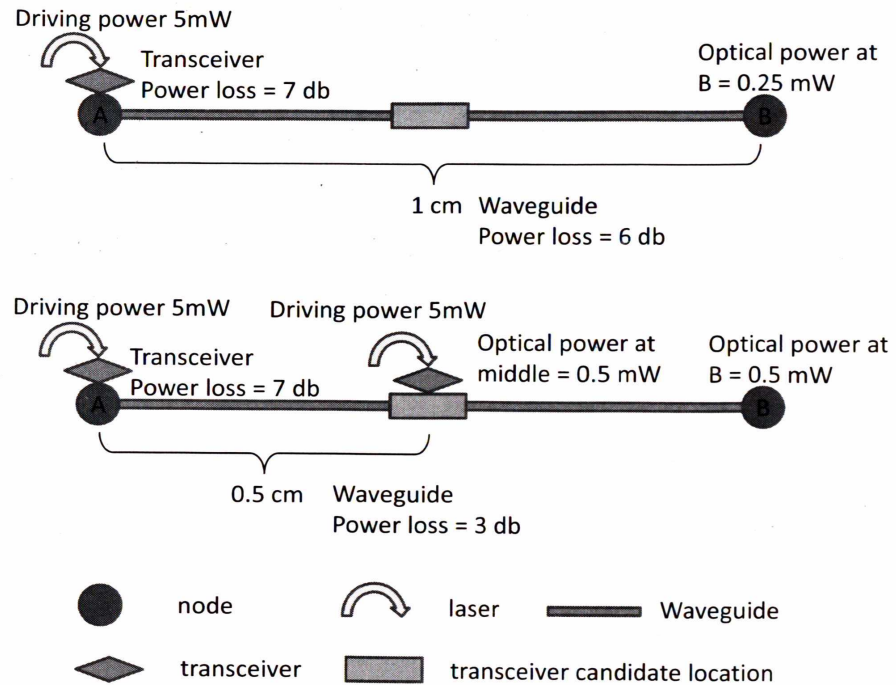


Figure 3.5: A transceiver insertion example.

technique and the one in [30] target for different scenarios.

3.2.4 Problem Formulation

Given an optical wire, denote photon-energy loss along this wire by its *energy demand*. Given a routing tree, we will create

some candidate transceiver locations in the tree such that the transceiver can be inserted there to meet energy demand. Refer to Figure 3.6 for a tree with the candidate transceiver location. We are also provided with a library of transceivers (which include transmitters), each of which has a driver with different size that may provide different amount of driving energy. We are to insert some transceivers at some candidate transceiver locations to meet the energy demand. On the other hand, as mentioned in Section 3.2.1, transceivers are also the sources of delay and cost. It is important to minimize their usage. In this project, we use $cost = \mu \cdot area + v \cdot power\ consumption$ to denote the cost of a transceiver.

In our optical integration, the two main tasks are: to construct a transceiver-aware optical routing curved tree and to insert

transceivers which is used to minimize the photon-energy loss under timing constraints. The transceiver-aware optical curved tree construction also needs to create the candidate transceiver locations. The problems are formulated as follows.

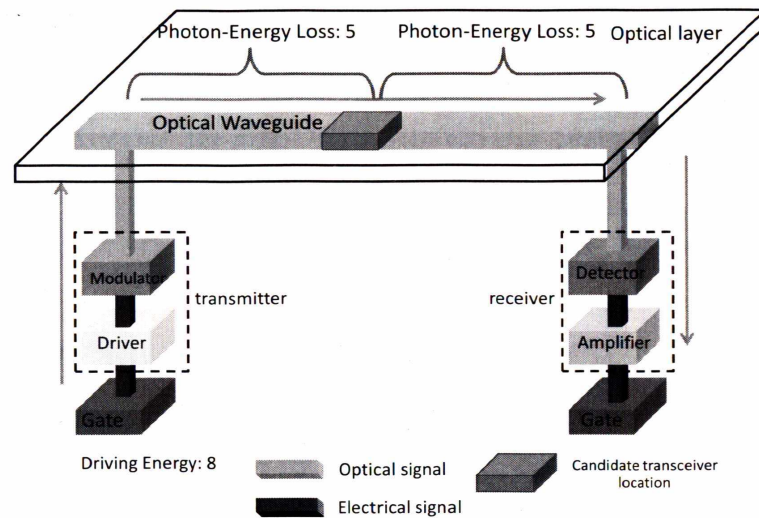


Figure 3.6: Illustration of a candidate transceiver location.

Transceiver-Aware Optical Curved Tree Construction

Problem: *Given a set of placed logic gates, to construct an optical curved tree to connect all the gates and to create*

candidate transceiver locations when necessary such that the total photon-energy loss over all edges is minimized.

Photon-Energy Driven Transceiver Insertion Problem:

Given a transceiver-aware optical curved tree with a set of candidate transceiver locations, and a set of available transceivers, to insert transceivers such that the energy demand and timing constraint are met and total cost overhead due to inserted transceivers is minimized.

Note that this work targets to provide a curved tree construction technique considering the optical energy loss. As the relationship between the curved tree construction and the optical interconnect routing is similar to that between the Steiner tree construction and global routing for copper

interconnects, our work can be viewed as a complementary effort to the pioneering work [30] which focuses on the routing for optical interconnects.

3.3 Photon-Energy Driven Transceiver-Aware Optical Curved Tree Construction

The first step is to compute a transceiver-aware optical curved tree construction. For this, an iterative algorithm is proposed which avoids the usage of couplers since couplers introduce extra photon-energy loss. During the algorithm, the candidate transceiver locations are also created.

The main feature for the algorithm is that our algorithm is transceiver insertion aware. That is, it will provide a solution

which is friendly to transceiver insertion. This is accomplished by involving transceiver-aware energy (which will be defined in Section 3.3.1) in forming the optical routing curved tree.

Given a set of driving gate and sink gates, the algorithm first constructs a *candidate curve* over every set of three gates. Note that a candidate curve could be either a curve or the combination of two line segments. Refer to Figure 3.7(a) for an example. After forming all candidate curves, our algorithm carries out an iterative procedure. In each iteration, the curve with minimum transceiver-aware energy is selected and added to the routing graph if this will not introduce any cycle. The process terminates when all the gates are processed (i.e., at least one curve involving the gate is added). The resulting selected gates may form some disconnected components, which will

then be connected with minimum length.

3.3.1 Curved Tree Construction

The algorithm first constructs a candidate curve over any set of three gates. In our implementation, a circle determined by the three gates is computed and the shorter portion of the circle containing all three gates is chosen as the candidate curve. Since in some three-gate set, connecting the gates using the curve leads to larger photon-energy loss than connecting them using the two line segments. When this happens, the combination of the two line segments (but not the part of the circle) is chosen to be the candidate curve. Refer to Figure 3.7(a) and Figure 3.7(b) for an example.

After forming the candidate curves, our algorithm carries out an iterative procedure to construct the routing tree graph. Initially, the routing graph is an empty graph. At each iteration, the algorithm greedily adds a candidate curve and the corresponding gates into the routing graph as long as there is no cycle in the graph. This process is iterated until all the gates are added in to the graph.

The greedy strategy precisely works as picking a candidate curve with minimum *transceiver-aware energy* to add. Let Cur denote a candidate curve. Given a curve Cur , define its *transceiver-aware energy*, denoted by $E(Cur)$, as follows.

$$E(Cur) = \alpha C(Cur) + \beta E(Cur), \quad (3.4)$$

where α, β are user-specified parameters, $E(Cur)$ denotes the photon-energy loss of Cur , and $C(Cur)$ denotes the estimated transceiver insertion cost (which refers to the combination of transceiver area and transceiver power consumption in this project). Since Eqn. (3.4) incorporates the (estimated) effect of transceiver insertion into consideration, our tree construction is transceiver insertion aware. To estimate $C(Cur)$, before the tree construction algorithm is performed, each candidate curve is treated as a simple tree and is sent to the procedure described in Section 3.3.2 for creating candidate transceiver locations. Suppose that there are k candidate transceiver locations along Cur returned by Section 3.3.2. $C(Cur)$ will be set to the product of k and the cost of the maximum cost transceiver.

Given a candidate curve with minimum transceiver-aware

energy, it will be added into the current routing graph as long as there would not result in any cycle in the graph. The purpose is to avoid the usage of couplers since a cycle necessitates a coupler. The following example illustrates this. As shown in Figure 3.8(a), assume a newly selected curve BDC leads to a cycle. Subsequently, the crossing at D needs a coupler as required by optical waveguide, which introduces extra energy. Instead of selecting BDC , the algorithm will use other candidate curve as shown in 3.8(b), which reduces photon-energy loss since photon-energy loss due to a coupler is typically large [2]. Although always avoiding coupler might lead to larger energy loss in certain cases, the above algorithm works well as demonstrated by our experiments.

The procedure can be formally described as follows. Let $G =$

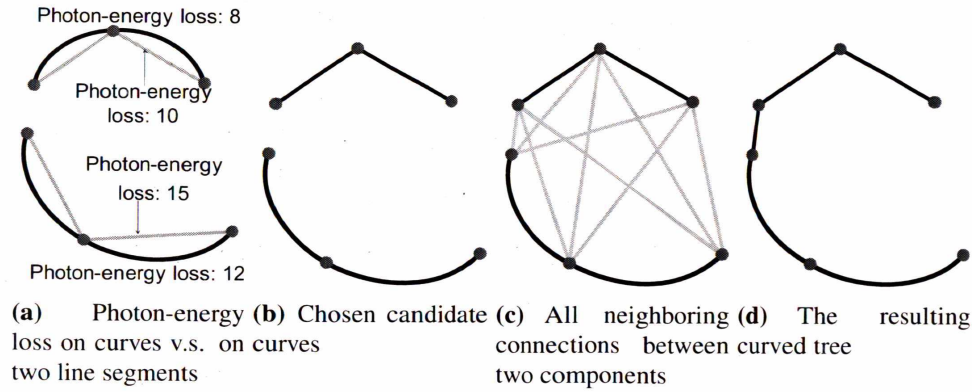


Figure 3.7: Optical Routing Procedures

$\{Cur_1, Cur_2, \dots\}$ denote the collection of all candidate curves.

Denote by T the routing tree graph. In the i -th iteration, a candidate curve will be added to T and some candidate curves will be removed out of G . Denote by G_i the G after the i -th iteration and by T_i the T after the i -th iteration, respectively. Initially, $G_0 = G$ and $T_0 = \emptyset$. In the i -th iteration, the candidate curve Cur_j for certain j with the minimum $E(Cur_j)$ in G_{i-1} is selected and removed from G_{i-1} , i.e., $G_i = G_{i-1} \setminus Cur_j$. Before Cur_j is added, the algorithm checks that once Cur_j is added,

whether there would be a cycle in T_i . If so, to avoid the coupler, Cur_j will not be added in T_i , i.e., $T_i = T_{i-1}$. Otherwise, $T_i = T_{i-1} \cup \{Cur_j\}$. This procedure is repeated until T_i contains all the gates or all $G_i = \emptyset$.

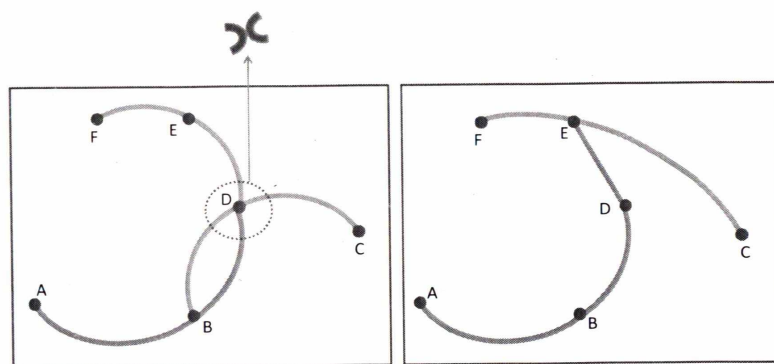


Figure 3.8: Tree construction avoiding coupler.

The resulting selected curves may form some disconnected components. The algorithm will then connect all the components using edge with minimum total length. That is,

for each disconnected component, the algorithm finds closest neighboring component, and connects to it using the minimum length wire segment. Note that the starting point and ending point of such a edge can be either a gate or a candidate transceiver location. After this, the whole photon-energy driven optical curved tree is constructed. Refer to Figure 3.7(c) for an illustration. Assume that there are two disconnected components in the routing graph. There are 9 possible paths between the two sets as shown as the orange lines. One just needs to greedily pick the shortest edge which results in Figure 3.7(d).

3.3.2 Creating Candidate Transceiver Locations

In an on-chip optical tree, the length of a curve may be long. Since the photon-energy provided by a driver in a transmitter is limited, it might not be able to provide enough driving energy. Therefore, a few candidate transceiver locations need to be set which can be used in the transceiver insertion later. This provides more flexibilities for the transceiver insertion in Section 3.4. Our candidate transceiver location setting ensures that there can be a transceiver insertion solution to satisfy the energy demand along every curve. Note that creating candidate transceiver locations is performed in the preprocessing stage, called by the process for estimating $C(Cur)$ as described in Section 3.3.1.

In the routing tree, given any two adjacent gates, we are to create some candidate transceiver locations as follows. Let c_p denote the curve between the two gates. There are a candidate transceiver location at each of the two gates since we need to connect the optical wire to each gate. We then assume that a typical transceiver is located at each of the two gates. Recall that $E(c_p)$ refers to the energy loss. Let E_{\max} denote the maximum driving energy that any transceiver can provide. If $E(c_p) \leq E_{\max}$, there is no need to create extra candidate transceiver location since a strong driver located at one end of c_p can provide sufficient energy to compensate for the energy loss. Otherwise, we will tentatively create a candidate transceiver location right in the middle of c_p . We again assume a typical transceiver inserted at this candidate

transceiver location. We split c_p into two parts and we can recursively apply the above process to each part. This process is recursed until $E(\cdot) \leq E_{\max}$.

3.4 Dynamic Programming Based Transceiver Insertion

A dynamic programming algorithm is proposed for photon-energy driven transceiver insertion. This process is somewhat similar to van Ginneken algorithm [42] and the algorithm in [43] for buffer insertion in classical copper wire based VLSI design. Our algorithm performs from the sink to the driver through the curved tree. In this proposed scheme, candidate transceiver insertion solutions Sol are generated from the sink to the driver. Denoted by S a transceiver insertion solution, $S \in Sol$, let tuple (E, C, T) to characterize the

insert solution, where E denotes the photon-energy loss from current node (a gate or a candidate transceiver location) to its immediately downstream transceiver/sink through the curved tree, C denotes the cumulated cost of the inserted transceiver in the subtree rooted at the current node, and T denotes required arrival time at the current node. Assume that the solution at a sink gate has $E = 0$, $C = 0$ and T is the required arrival time at the sink. Note that multiple transceivers with different sizes are available. The proposed algorithm includes the procedures of “add optical wire”, “add transceiver”, “merge branch” and “pruning” during the solution propagation. After a solution is propagated to the driving gate, the solution with the minimum cost C among all the solutions satisfying timing and energy demand is chosen to be our solution.

3.4.1 Add Optical Wire

For a upstream node u with no branching point in between, the solution S_v can be propagated directly. If no transceiver is inserted, only the optical routing needs to be considered.

Denote by $e_{u,v}$ the curve between node u and node v . $E(e_{u,v})$ gives the energy loss along the edge and $T(e_{u,v})$ denotes the timing degradation along the edge. The updated solution S_u at node u can be generated as follows.

$$\begin{aligned}
E(S_u) &= E(S_v) + E(e_{u,v}), \\
C(S_u) &= C(S_v), \\
T(S_u) &= T(S_v) - T(e_{u,v}),
\end{aligned}
\tag{3.5}$$

where $E(e_{u,v})$ can be computed by Eqn. (3.1) and Eqn. (3.2).

The optical wire delay $T(e_{u,v})$ is linearly with the length and unit wire delay.

3.4.2 Add Transceiver

Assume that a transceiver, denoted by κ , is added at node u to provide driving energy. This is valid if κ can provide the driving energy at least $E(S_u)$. Otherwise, a transceiver with

large driving energy has to be used. If there is no transceiver which can provide the driving energy $\geq E(S_u)$, the solution is eliminated from the consideration. Suppose a transceiver is inserted at u . Viewing from its input, the energy demand becomes zero. The solution S_u at node u will be updated as solution S'_u as follows.

$$\begin{aligned}
E(S'_u) &= 0, \\
C(S'_u) &= C(S_u) + C(\kappa), \\
T(S'_u) &= T(S_u) - T(\kappa),
\end{aligned}
\tag{3.6}$$

where $T(\kappa)$ is the delay of the transceiver κ and $C(\kappa)$ is the cost of the transceiver κ . As mentioned in Section 3.2, there are a set of available drivers with different driving energies. Each different driving energy corresponds to a different $C(\kappa)$ and results in one solution in the transceiver insertion.

3.4.3 Merge Branch

Solutions has to be merged at the merging point of branches. Let B_1 and B_2 denote the two branches. The solutions at both branches are already propagated to (precisely, the nodes immediately downstream to) the branching node. One needs to guarantee that the energy demand is met in each branch. If only one transceiver is inserted at the branching node, it is difficult to guarantee that the driving energy can be appropriately

distributed to compensate for the photon-energy loss to each branch. Refer to Figure 3.9 for an example. If one transceiver with the 12-unit driving energy is inserted at node u which is immediately upstream to the branching node, there is no guarantee that 6-unit energy will be distributed to B_1 and the rest 6-unit energy will be distributed to B_2 . To handle this issue, a conservative strategy is used. That is, each branch will be associated with a separate transceiver immediately downstream to the branching node. In this case, the cumulated energy loss along wire is set to zero as in “add transceiver” case. On the other hand, the required arrival time is set to the less one of the two branches. Formally, to merge two solutions at branching node u , two transceivers κ_1 and κ_2 are inserted at the locations

immediately downstream to the branching node u , and we have

$$E(S_u) = 0$$

$$C(S_u) = C(S_{u,B_1}) + C(S_{u,B_2}), \quad (3.7)$$

$$T(S_u) = \min\{T(S_{u,B_1}), T(S_{u,B_2})\},$$

where $C(S_{u,B_1})$ (resp. $C(S_{u,B_2})$) and $T(S_{u,B_1})$ (resp. $T(S_{u,B_2})$) are the cumulated cost and the required arrival time at the node immediately downstream to u along branch B_1 (resp. B_2).

3.4.4 Pruning

The pruning technique is performed to accelerate the above procedure and reduce the number of solutions. Recall that a solution is pruned if no transceiver can provide the sufficient

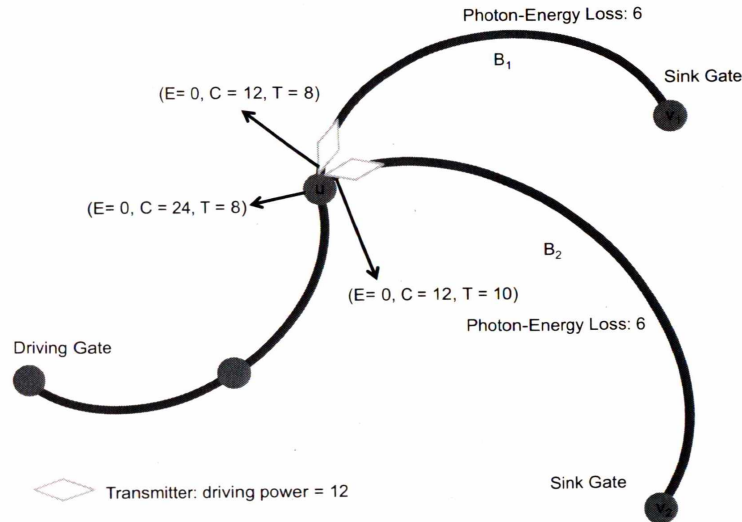


Figure 3.9: Illustration of branch merge.

amount of driving energy needed from the downstream part as described in “add transceiver”. When a solution has a arrival time of which is less than the driver requiring arrival time, it will also be pruned since it cannot satisfy the timing constraint.

An example shown in Figure 3.10 illustrates the transceiver insertion procedure. Assume that at the node D (sink), $E = 0$, $C = 0$ and $T = 80$, and the arrival time at the node A (driver) is

0. After adding the optical wire between node C and node D , the solution is $S_C = (6, 0, 70)$. At node C , if a transceiver κ_1 is inserted, the solution is $S'_C = (0, 40, 50)$. If a transceiver κ_2 is inserted, the solution is $S''_C = (0, 10, 65)$. These three solutions are propagated to node B through adding the optical wire between B and C . Among the three updated solutions, it is easy to see that the solution propagated from S'_C is $(36, 40, -10)$. Since this solution violates the timing constraint, it is pruned. After pruning, the remaining solutions will be processed for transceiver insertion and further propagated to the driver.

3.5 Experimental Results

The proposed photon-energy driven transceiver-aware optical curved tree construction algorithm and the dynamic

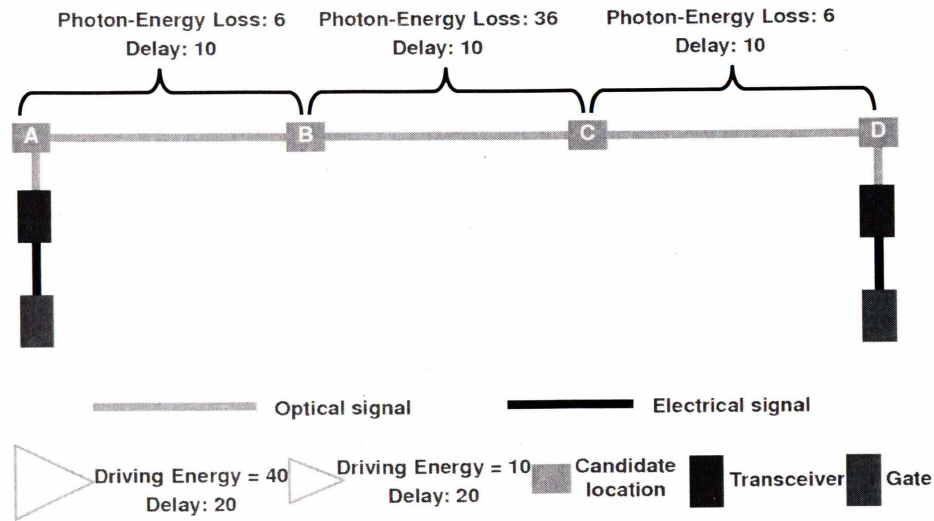


Figure 3.10: Illustration of transceiver insertion.

programming based transceiver insertion algorithm. Since the parameters in [30] are not available to us, we do not compare to [30]. In fact, these two techniques target for different scenarios. We conduct the experiments on a set of 500 nets and compare our curved tree construction algorithm (denoted by NEW) to a natural minimum spanning tree heuristic (denoted by MST). In this work, the 32nm technology is used and the parameters are

from [2]. The MST heuristic constructs a minimum spanning tree over the gates using the line segments but not the curves. We also compare the performance of the novel optical trees with that of the conventional copper trees (denoted by Copper). The copper trees are from [43] which are extracted from an industrial ASIC chip, and we scale them to $32nm$ technology node followed by buffer insertion. When the optical tree of NEW (resp. MST) is constructed, the copper tree is ripped up and rerouted using NEW (resp. MST). Since transceiver insertion is performed on optical curved trees, a transceiver library is needed. Due to that there is no industrial transceiver library, we generate ten types of transceivers where the mean value of power consumption is 6.2 mW according to [2] and the mean value of modulator area is $5\text{ }\mu m^2$ according to [39].

The power and area for the transceivers are shown in Table 3.3.

Table 3.3
The power and area of each transceiver.

Transceiver	Power (<i>mW</i>)	Area (<i>um</i> ²)
Transceiver 1	5.2	2.5
Transceiver 2	5.4	3
Transceiver 3	5.6	3.5
Transceiver 4	5.8	4
Transceiver 5	6.0	4.5
Transceiver 6	6.2	5
Transceiver 7	6.4	5.5
Transceiver 8	6.6	6
Transceiver 9	6.8	6.5
Transceiver 10	7.0	7

The comparison among different tree constructions is summarized in Table 3.4 in which the results are averaged over 500 nets. Timing refers to the circuit delay. CPU refers to the runtime in seconds. The parameters from [2] are used. It shows the results averaged over 500 nets. In timing evaluation, the predictive delay parameters for optical interconnects (as shown in Table 3.1 and Table 3.2) and those for copper interconnects

are from [2]. We make the following observations.

- † One can see that the circuit timing of Copper is as large as 198ps, while MST has the timing of only 43ps. This means that optical routing can significantly reduce the delay.
- † NEW introduces some detour compared to MST for optimizing photon-energy loss and its timing degrades to 145. However, this is still much better than that of Copper.
- † The main purpose of NEW is to reduce photon-energy loss. The actual effect can be clearly seen from Table 3.4. Compared to MST, the energy loss reduces from 224 to 97 which is quite significant.
- † NEW and MST are as efficient as each other. It seems that more computation effort is needed in NEW compared

to MST. However, due to the fact that most nets are very small (with just a few sinks), NEW actually runs very fast.

Table 3.4
Comparison of tree constructions.

	Energy Loss	Timing (ps)	CPU (s)
Copper	–	198	–
MST	224	43	0.01
NEW	97	67	0.01

We next perform transceiver insertion to the constructed curved tree. If we do not create any extra candidate transceiver locations in middle of edges, we observe that 55.2% of nets still cannot meet energy demand even after transceiver insertion. Thus, for those nets, it is necessary to apply our technique in Section 3.3.2 to create candidate transceiver locations. Later, transceiver insertion can be performed to meet energy demand on every wire.

Table 3.5 summarizes the averaged result after transceiver insertion to compensate for the energy loss on all 500 nets. In this case, we require that the energy demand in each edge of the optical tree is satisfied. Our algorithm can achieve this. Transceiver cost is the combination of transceiver area and transceiver power consumption due to the inserted transceivers. Note that we use the same procedure to create candidate transceiver locations and perform transceiver insertion for MST and NEW. We make the following observations.

† After transceiver insertion, both routing trees satisfy the energy demand.

† The transceiver cost by MST is 65 which is $1.5\times$ larger than that of NEW which has cost of only 43. This

is because NEW has much smaller power loss. It demonstrates the effectiveness of our transceiver aware tree construction.

† Without creating the candidate transceiver locations, the energy demand cannot be satisfied in 46.5% of the 500 nets. Thus, our algorithms are highly necessary.

† Transceiver insertion can be performed very efficiently. The results on both MST and NEW show that our transceiver insertion takes only about 0.60 seconds per net.

† Although the timing of NEW is somewhat worse than that of MST, it is acceptable since the transceiver cost by NEW is much better.

We also show the results of five selected nets in Table 3.6 in

Table 3.5
Comparison results after transceiver insertion.

	Transceiver Cost	Timing (ps)	CPU (s)
MST w/ trans.	65	142	0.58
NEW w/ trans.	43	133	0.60

Table 3.6
Results on five selected nets which need to create the extra candidate transceiver locations

Net	Energy Loss	Transmission Violation		Transceiver Cost			Timing (ps)			CPU (s)		
		After T.I. w/o extra	After T.I. w/ extra	Before T.I.	After T.I. w/o extra	After T.I. w/ extra	Before T.I.	After T.I. w/o extra	After T.I. w/ extra	Before T.I.	After T.I. w/o extra	After T.I. w/ extra
1	266	Yes	No	0	45	120	196	264	377	0.01	0.20	0.22
2	278	Yes	No	0	45	90	216	284	351	0.01	0.26	0.27
3	379	Yes	No	0	59	132	141	321	344	0.01	2.58	2.61
4	255	Yes	No	0	59	88	215	306	350	0.01	2.81	2.89
5	338	Yes	No	0	59	118	197	288	378	0.01	2.65	2.70

which T.I. refers to transceiver insertion. "Before T.I." refers to the result of tree construction. "After T.I. w/o extra" refers to the result of applying transceiver insertion algorithm to the tree without any extra candidate transceiver locations in middle of any edge. Since the selected nets shown in Table 3.6 all need the extra candidate transceiver locations, none of them can meet energy demand (i.e., there is at least one edge whose

driver cannot provide sufficient driving energy to compensate for the photon-energy loss along the edge). () in “After T.I. w/o extra loc.” and “After T.I. w/ extra loc.” specifies the status of transmission violation. “No” means that there is at least one edge of this net with violations where the driver cannot provide sufficient driving energy to compensate for the photon-energy loss along the edge. In contrast, “Yes” means that the optical transmission succeeds. Table 3.6 clearly demonstrates the need to apply our transceiver insertion framework for meeting the energy demand.

3.6 Conclusion

In this project, the first transceiver insertion technique for on-chip optical integration is proposed. It includes an efficient

dynamic programming based transceiver insertion algorithm and a novel transceiver-aware tree construction algorithm. Experimental results on 500 timing critical nets demonstrate that our algorithm can reduce the energy demand by $1.9\times$ compared to a natural minimum spanning tree heuristic, and our constructed optical trees can improve the timing by about $2\times$ compared to copper trees. The transceiver insertion algorithm can always satisfy the energy demand while still minimizing transceiver cost. In fact, without transceiver insertion, 55.2% nets cannot meet the photon-energy demand.

Chapter 4

Fault-Aware Energy Efficient Parallel Task Scheduling¹

4.1 Introduction

The transient faults number in embedded system increases due to the increasing level of integration and reducing size of transistor features in addition to harsh operating environments

¹©2011 IEEE. Portions reprinted with permission, from Tongquan Wei, **Xiaodao Chen** and Shiyan Hu, “*Reliability-Driven Energy-Efficient Task Scheduling for Multiprocessor Real-Time Systems*”, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 30, pp. 1569-1573, 2011.

[6]. The probability of fault occurrences is even higher in a multiprocessor system as a result of large number of components and increased design complexity. Since real-time applications demand both temporal and logical correctness, it is desirable that in the presence of faults real-time tasks finish execution before their respective deadlines. Therefore, real-time embedded systems are typically designed with enough margins to tolerate the worst case expected number of faults by trading off fault coverage and fault detection latency with system performance. Fault tolerance of real-time multiprocessor systems is typically achieved through primary-backup (PB) approach[44, 45, 46].

The need for energy-efficient design is increasing for battery-powered real-time systems to reduce power density and

enhance the system operational lifetime. Dynamic Voltage Scaling (DVS) is a popular system level power management technique that exploits technological advances in power supply circuits to reduce processor power consumption by dynamically scaling down the processor speed. Numerous task allocation and scheduling techniques based on DVS have been proposed for energy minimization in multiprocessor systems [47, 48, 49, 50, 51, 52]. However, using DVS technique to reduce power consumption has a negative effects on system reliability. It has been shown that scaling down the processor speed increases the transient faults rates, especially for those induced by cosmic ray radiations, and thus degrades system reliability [53].

Rollback recovery and re-execution technique are typically

utilized to handle transient faults [54]. Task schedule also can use slack time to slow down processor speed. When more slack is used for fault recovery through rollback execution, less slack is left for energy savings via voltage scaling. Therefore, fault-tolerance and energy are two design constraints that interplay and need to be jointly optimized.

The joint optimization of energy and fault-tolerance as two important design constraints for safety-critical real-time systems has been extensively investigated in the recent past [55, 56, 57, 58, 59, 60]. Both energy savings and fault-tolerance are achieved by utilizing the slack time in a task schedule. However, all these researches focus on the joint optimization of the two design constraints for uni-processor systems. Wei et. al [61] proposed an energy driven task scheduling schemes

with deterministic fault-tolerance capabilities for symmetric multiprocessor systems executing tasks with hard real-time constraints. Without considering the faults and application timing requirements, optimum energy savings can be achieved. However, the presented scheme can not handle the scheduling of tasks with precedence constraints. In [62], the authors described a flexible multiprocessor platform using modest hardware support to enable an energy efficient fault-tolerance mechanism. Timeliness of the system is not considered as a design constraint. In [63], the authors addressed the scheduling and voltage scaling for hard real-time applications that have been statically mapped on heterogeneous distributed embedded systems. Tasks are assumed to share one deadline and the effect of voltage scaling on system reliability is

taken into account. Similar to [63], the authors of [64] also investigate the reliability aware power management for real-time tasks sharing a common deadline, but they assume the investigated multiprocessor system is homogenous and tasks to be scheduled are independent.

This project proposes a reliability-driven energy efficient task scheduling scheme for real-time homogeneous or heterogeneous multiprocessor embedded systems that optimizes system energy consumption under stochastic fault occurrences. The proposed multiprocessor task scheduling scheme is featured by a novel fault adaptation variable β that models the uncertainties in fault occurrences. For a given target reliability, the proposed scheme accelerates the computing of the desired task schedule by utilizing techniques such as the

β -enabled parallel scheduling, sequential rounding, and Latin Hypercube sampling-based Monte Carlo simulation. The main contributions of the project are summarized as follows.

† The proposed scheme considers both the dynamic power and the leakage power, is able to handle the scheduling of both independent tasks and tasks with precedence constraints, and is capable of scheduling tasks with varying deadlines.

† The task allocation and scheduling is formulated as an Integer Linear Programming. Several techniques such as the sequential rounding and Latin Hypercube sampling-based Monte Carlo Simulation are adopted to speed up the computation of the desired task schedule.

† The fault adaptation variable β models the uncertainties of fault occurrences and enables the scheduling parallelism of the proposed scheme. The energy consumption is minimized under the stochastic fault occurrences.

† Simulation results have shown that the proposed scheme saves energy more than 15% by comparing to the approach of designing for the corner case of fault occurrences, and exhibits a speedup of up to $3.1\times$ on a machine with Intel Core 2 Quad processor.

4.2 System Architecture and Models

Focused on the study is on multiprocessor systems, this project considers the inter-core communication is implemented by shared memory. As a result, the cost for inter-core

communication can be assumed to be negligible. It is also assumed that each processor unit has DVS ability and supports limited number of discrete frequency levels.

4.2.1 Energy Model

The power consumption consists of the static power consumption and the dynamic power consumption. The static power consumption is given by

$$p_s = V_{dd}I_{subn} + |V_{bs}|I_j,$$

where V_{dd} denotes the supply voltage, I_{subn} denotes the sub-threshold leakage current, V_{bs} denotes the bias voltage, I_{subn} is the sub-threshold leakage current, and I_j denotes the reverse bias junction current [65]. Since and the V_{bs} and I_j are

technology constants, the static power can be characterized as a function of the power supply and is fixed for a certain supply voltage.

The dynamic power consumption is given by $p_d \propto V_{dd}^2 f$, where V_{dd} and f denote the voltage and the processor frequency respectively. The operating frequency can hence be approximated as a linear function respect to the voltage [66].

As a result, the dynamic power consumption can be estimated by a convex increasing function, that is, $p_d \propto f^3$.

The energy consumption of a CMOS device can be reduced by decreasing the dynamic or static energy consumption of the device. The dynamic energy consumption is reduced by using DVS technique to reduce the CPU speed. However, scaling

down the processor speed prolongs the interval over which the computation is carried out; thus, increases static energy consumption. The static energy consumption can be reduced by turning off the processor after task execution completed. However, this strategy of reducing static energy consumption leads to an increase in dynamic energy consumption. Hence, the reduction in the dynamic energy and static energy needs to be judiciously balanced to minimize the total energy consumption. The total energy (E_{tot}) consumed by a multiprocessor system during a computation interval $[t_1, t_2]$ is estimated as follow

$$E_{tot} \propto K \times (t_2 - t_1) \times (p_d + p_s), \quad (4.1)$$

where K denotes the total number of CPUs, and p_d and p_s are dynamic and static power consumption, respectively.

4.2.2 Model for Fault Estimation

Poisson distribution is used to model transient faults, in this way during an interval T , the probability of k faults is formulated as follow

$$\frac{e^{-\lambda T} (\lambda T)^k}{k!}, \quad (4.2)$$

where λ is the average transient fault arrival rate and λT is the average number of fault occurrences during the interval T . The transient faults rate is highly related to processor frequency and

supply voltage, and the fault arrival rate of a single chip can be approximated by

$$\lambda = \gamma \times e^{-\alpha f}, \quad (4.3)$$

where γ and α are constant parameters, and f is the current operating frequency [53]. It can be deduced from Equation (4.3). Note that the fault arrival rate in average is fixed for a given frequency, and increases with the scaling down of the frequency.

This project is based on a watchdog processor with timing checking strategy. Fault detection is based on checkpointing roll back recovery method [6]. The checkpointing overhead is

denoted by c_s , which is often deemed to be constant.

4.2.3 Application and Recovery Model

It is assumed that the target multiprocessor system consists of N processing units and each processing element supports L discrete frequency levels. Consider a real-time task set Γ consisting of M periodic tasks with precedence constraints: $\{\Gamma | \tau_1, \tau_2, \dots, \tau_M\}$. A task is ready for execution only if the execution of its predecessor is completed. Similarly, the successor of a task is ready for execution only if the execution of the task is completed. To characterize a task τ_m is defined as a tuple, $\tau_m = \{T_m, D_m, C_m\}$, where T_m and D_m are the period and the deadline respectively, and C_m denotes the task execution cycles. We assume that, $T_m = D_m$. Each task is assigned to one

and only one processor, and the task executes at one and only one frequency level. The later two assumptions can be relaxed and the proposed technique is still applicable if the frequency switching overhead during the execution of a task and the task migration overhead among processors are carefully accounted in task execution times.

Checkpointing technique is used in this work to provide fault-tolerance. It is assumed that checkpointing intervals for a given task are equal. Let k_m be the worst case number of fault occurrences of task τ_m at the frequency level l , f_m be the operating frequency of task τ_m at the frequency level l , and O_m be the optimal number of checkpoints for task τ_m that minimizes the task execution time at the frequency level l , then

O_m is given by

$$O_m = \left\lceil \sqrt{\frac{k_m C_m}{c_s f_m}} - 1 \right\rceil \quad \text{or} \quad \left\lfloor \sqrt{\frac{k_m C_m}{c_s f_m}} - 1 \right\rfloor, \quad (4.4)$$

where c_s is the checkpointing overhead [57]. Assuming the best case where no faults occur during the execution of task τ_m , then CB_m , which is defined to be the execution cycles of task τ_m with only checkpointing overhead, is expressed as

$$CB_m = C_m + O_m \times c_s \times f_m. \quad (4.5)$$

Assuming the worst case where faults occur at the end of checkpointing saving, then CW_m , which is defined to be the execution cycles of task τ_m including checkpointing overhead

and the worst case error recovery overhead, is given by

$$CW_m = CB_m + \frac{k_m C_m}{(O_m + 1)} + 2k_m \times c_s \times f_m. \quad (4.6)$$

In Equation (4.6), $\frac{C_m}{(O_m + 1)}$ denotes the checkpoint interval and $\frac{k_m C_m}{(O_m + 1)}$ indicates the overhead to recover from k_m fault occurrences. The term $2k_m \times c_s \times f_m$ gives the overheads of k_m checkpoint savings and k_m system state retrievals [57].

CB_m and CW_m of task τ_m are constant for a fixed operating frequency f_m or frequency level l of task τ_m . However, the current execution time C_m of task τ_m is a random variable due to the stochastic property of fault occurrences. Since it is difficult to solve a mathematical program with uncertainty, the energy optimization problem is transformed into a deterministic optimization problem without random

variables. In this work, a variable β , referred to as fault adaptation variable, is introduced to model the uncertainty in task execution time due to fault occurrences. The current execution time of task τ_m including fault recovery overhead can be expressed as a function of CB_m and CW_m , that is,

$$C_m = (1 - \beta) \times CB_m + \beta \times CW_m, \quad (4.7)$$

where $0 \leq \beta \leq 1$. The execution time of task τ_m is CW_m if $\beta = 1$ and is CB_m if $\beta = 0$.

Let λ_l denote the average fault arrival rate at the frequency level l for $1 \leq l \leq L$, where L is the number of processor supported frequency levels. Given the execution time $\frac{C_m}{f_m}$ of task τ_m at the frequency f_m , the average number of fault occurrences during

the task execution can be written as

$$\lambda_l \frac{C_m}{f_m},$$

where λ_l at frequency f_m can be calculated by the Equation (4.3). The fault probability k_m at frequency f_m of task τ_m can be deduced from Equation (4.2), which is re-written as

$$\frac{e^{-\lambda_l \frac{C_m}{f_m}} \times (\lambda_l \frac{C_m}{f_m})^{k_m}}{k_m!}. \quad (4.8)$$

The reliability of a task is the probability of completing the task successfully subject to faults [53]. As a result, the reliability of task τ_m is the probability of completing the task successfully subject up to k_m faults. The task level reliability is maintained if all tasks in the task set finish the execution successfully under their respective given reliability target. Let RG_m denote the

reliability goal of task τ_m and R_m denote the reliability of the task, the reliability of task τ_m is maintained if the inequality

$$RG_m \leq R_m = \sum_{k_m} \frac{e^{-\lambda_l \frac{C_m}{f_m}} \times (\lambda_l \frac{C_m}{f_m})^{k_m}}{k_m!} \quad (4.9)$$

holds for $\beta = 1$. Since λ_l, C_m, f_m , and k_m all are functions of the frequency level l of task τ_m , R_m is also a function of the frequency level l of task τ_m . For a given frequency level l ($1 \leq l \leq L$) and $\beta = 1$, λ_l, C_m , and f_m are all known; thus, the worst case number of fault occurrences k_m subject to target reliability RG_m can be iteratively derived using the inequality (4.9).

4.3 Integer Linear Program (ILP) Formulation

The reliability-driven energy efficient task-to-processor assignment and scheduling is formulated as an integer linear

programming (ILP) problem. The object function is the multiprocessor energy consumption that considers both the dynamic power and leakage power. According to energy consumption estimation module given in Equation (4.1) and the task execution time given in Equation (4.7), the energy consumption of task τ_m with execution time $\frac{C_m}{f_m}$ is expressed as $\frac{C_m}{f_m}(p_d + p_s)$, and the total energy consumption of a given task set, E_{tot} , can be derived by adding together energy consumptions of all tasks in the task set, that is,

$$E_{tot} = \sum_{m=1}^M \frac{C_m}{f_m} (p_d + p_s). \quad (4.10)$$

Energy optimization is performed under constraints of the task execution time as a function of the fault adaptation variable β for both independent and dependent tasks with varying

deadlines. In this section, the formulation for independent tasks with a common deadline is first given. Then it is extended to tackle the scheduling of both independent and dependent tasks with varying deadlines.

In the integer linear program definition given below, the variable $A_{m,l,n}$ denotes the scheduling of task m at the frequency level l on processor n . $A_{m,l,n}$ is equal to 1 if task τ_m is scheduled at the l -th frequency level on processor n , and is equal to 0 if τ_m is scheduled at any other frequency levels or on any other processors. Given independent tasks τ_m for $m = 1, 2, \dots, M$ with the same deadline of D , the number of processors N , the number of processor supported frequency levels L , and a value of β for $0 \leq \beta \leq 1$, the goal is to find $A_{m,l,n}$ for $m = 1, 2, \dots, M$, $l = 1, 2, \dots, L$, and $n = 1, 2, \dots, N$, that satisfy

the linear constraints

$$A_{m,l,n} = 0, 1$$

$$\sum_{l=1}^L \sum_{n=1}^N A_{m,l,n} = 1$$

$$\sum_{m=1}^M \left(\frac{C_m}{f_m} \right) A_{m,l,n} \leq D$$

and minimize the multiprocessor energy consumption $E_{tot}(A_{m,l,n})$ given in Equation (4.10).

This ILP definition can be extended to tackle the scheduling of both independent and dependent tasks with varying deadlines. For the sake of easy presentation, an ILP definition for single processor task scheduling is described below to demonstrate the technique. Let $A_{m,l}$ denote the scheduling of task τ_m at the frequency level l . $A_{m,l}$ is equal to 1 if task τ_m is scheduled at

the l -th frequency level, and is equal to 0 if τ_m is scheduled at any other frequency levels. Let the variable S_i and S_j denote the start time of task τ_i and task τ_j , respectively, and the variable $Smin_{(i,j)}$ denote the minimum of the S_i and S_j . Thus, the equation $Smin_{(i,j)} = min(S_i, S_j)$ holds for any two tasks in a given task set. The $b_{i,j}$ is a auxiliary binary decision variable indicating the relationship of $Smin_{(i,j)}$, S_i , and S_j . If $S_i < S_j$ holds, that is, $Smin_{(i,j)} = S_i$, then $b_{i,j} = 1$; else $b_{i,j} = 0$. H is a large constant number and is set to 10000 in the experimental section. Inequality (4.13) to Equation (4.17) ensure that $Smin_{(i,j)} = min(S_i, S_j)$ holds. Similar to $Smin_{(i,j)}$, the variable $Smax_{(i,j)}$ is introduced to indicate the maximum of the S_i and S_j . That is, $Smax_{(i,j)} = max(S_i, S_j)$ holds for any two tasks in a given task set. Two auxiliary variables $h_{i,j}$ and $g_{i,j}$

are also introduced as pseudo-linear constraints to facilitate the formulation .

Given independent or dependent tasks τ_m with respective deadlines of D_m for $m = 1, 2, \dots, M$, the number of processor supported frequency levels L , and a value of β for $0 \leq \beta \leq 1$, the goal is to find $A_{m,l}$ and S_m for $m = 1, 2, \dots, M$ and $l = 1, 2, \dots, L$ that minimize the system energy consumption under the given constraints. The formulation of the ILP problem is

thus given as follows.

$$\text{minimize: } E_{tot}(A_{m,l}; S_m)$$

$$\text{subject to: } A_{m,l} = 0, 1 \quad (4.11)$$

$$\sum_{l=1}^L \sum_{n=1}^N A_{m,l} = 1 \quad (4.12)$$

$$S_{min(i,j)} \leq S_i \quad (4.13)$$

$$S_{min(i,j)} \leq S_j \quad (4.14)$$

$$S_{min(i,j)} \geq S_i - H \times (1 - b_{i,j}) \quad (4.15)$$

$$S_{min(i,j)} \geq S_j - H \times b_{i,j} \quad (4.16)$$

$$b_{i,j} = 1, 0 \quad (4.17)$$

$$S_{max(i,j)} = S_i + S_j - S_{min(i,j)} \quad (4.18)$$

$$h_{i,j} = H \times (S_i - S_{max(i,j)}) + S_j \quad (4.19)$$

$$S_i - h_{i,j} \geq \sum_{l=1}^L A_{j,l} \left(\frac{C_j}{f_j} \right) \quad (4.20)$$

$$g_{i,j} = H \times (S_j - S_{\max(i,j)}) + S_i \quad (4.21)$$

$$S_j - g_{i,j} \geq \sum_{l=1}^L A_{i,l} \left(\frac{C_i}{f_i} \right) \quad (4.22)$$

$$S_i + \sum_{l=1}^L A_{i,l} \left(\frac{C_i}{f_i} \right) \leq D_i \quad (4.23)$$

$$S_{m_1} < S_{m_2} \quad (4.24)$$

$$(\tau_{m_1}, \tau_{m_2}) \in \Gamma; 1 \leq l \leq L$$

Equation (4.11) restates the definition of the variable $A_{m,l}$. That is, $A_{m,l}$ is either 0 or 1. Equation (4.12) indicates that each task runs at one and only one processor frequency level. Inequality (4.13) to Equation (4.17) is the constraint formulation for the equation of $S_{\min(i,j)} = \min(S_i, S_j)$. Equation (4.18) identifies

the maximum of S_i and S_j . Equation (4.19) to (4.22) ensure that the executions of task τ_i and τ_j on the processor have no overlapping. The execution cycles C_i of task τ_i can be derived using the Equation (4.7). Equation (4.23) indicates that each task has to finish the execution before its deadline. Equation (4.24) enforces the precedence constraints assuming task τ_{m_2} can not start until the execution of task τ_{m_1} finishes. This formulation can be easily extended to handle the scheduling of tasks on multiple processors. Due to space limitation, it is not presented in this project.

4.4 Reliability-Driven Parallel Task Scheduling for Multiprocessor Systems

Different requirements of safety in various applications necessitates the reliability-driven task scheduling for multiprocessor embedded systems. The reliability-driven energy efficient task scheduling aims to generate an energy optimum schedule and meet the target task level reliability goal under the assumption of the Poisson probability distribution of fault occurrences.

A systematic reliability-driven parallel task scheduling algorithm referred to as RDPS, is proposed in this section to derive the desired fault adaptation variable β and generate the energy optimum task schedule for a given task set with

fault-tolerance requirements. As described in Fig. 5.3, for a given task set Γ and the target reliability RG_m , a value of the fault adaptation variable β is randomly picked, and the associated fault recovery overhead is incorporated in task execution time based on the selected β (step A). The schedule of the task set is then generated by solving the integer linear program definition (step B). Finally, the reliability R_m of each task is derived using Monte Carlo simulation (step C). If the reliability of the a task does not satisfies the stop-condition of the RDPS algorithm, the fault adaptation variable β is adjusted and the above process is repeated. If the reliability of all tasks satisfies the stop-condition of the RDPS algorithm, the output task schedule is optimal in energy consumption and its reliability meets the system reliability requirement. In other

Require: task set Γ & the reliability goal RG_m

Ensure: the desired β & energy optimum task schedule

1. randomly pick a value of β in $0 \leq \beta \leq 1$
2. **repeat**
3. A: update C_m of τ_m based on β , then update β
4. B: generate a task schedule by solving the ILP
5. C: derive R_m of task τ_m using Monte Carlo simulation
6. **until** $(R_m - RG_m) \geq \varepsilon > 0$

Figure 4.1: Reliability-driven parallel task scheduling (RDPS)

words, the output task schedule is the desired task schedule if $(R_m - RG_m) \geq \varepsilon > 0$ holds, where ε is a arbitrarily small positive number. The following subsections describe each step of the RDPS algorithm in details.

4.4.1 β -Enabled Parallel Multiprocessor Task Scheduling

One of the key contributions of this work is to introduce a fault adaptation variable β that adapts task execution time including fault recovery overhead to the Poisson probability distribution of fault occurrences. Unlike the traditional approach of

designing for corner cases, this novel technique enables the designing of reliability-driven energy efficient real-time embedded systems based on status quo of fault occurrences. Since $0 \leq \beta \leq 1$, the execution time of task τ_m ranges from CB_m to CW_m , as is shown in Equation (4.7).

Due to the statistical property of transient fault occurrences, there exists no deterministic relationship between the fault adaptation variable β and the reliability of a task. As a result, the current reliability R_m of task τ_m , if does not meet the stop-condition of the RDPS algorithm, can not be used to direct the selection of the value of β to be used in the next iteration of the RDPS algorithm. A simple yet efficient binary search-based approach is hence utilized to obtain the next values of β . More specifically, given the initial range of $\beta \in [0, 1]$, and the initial

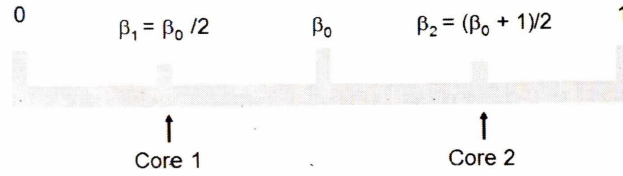


Figure 4.2: Select the next candidates of β using the binary-search algorithm

value of β denoted by β_0 , the next values of β could be in the range of both $[0, \beta_0]$ and $[\beta_0, 1]$. If the current reliability R_m of task τ_m does not reach the specified reliability goal of RG_m , the next candidates of β that will be used to generate the task schedule are $\beta_1 = \beta_0/2$ and $\beta_2 = (\beta_0 + 1)/2$, as is illustrated in Fig. 4.2. Repeat this process until the R_m is greater than yet close enough to the RG_m .

The introduction of the fault adaptation variable β also enables the parallel computing of task schedules by utilizing the powerful computing capacity of modern computers. The

iterative RDPS task scheduling algorithm shown in Fig. 5.3 is essentially a pipelined processing stages, each of which consists of using β to adapt task execution time to fault occurrences, solving linear program to generate a task schedule, and evaluating the resultant task schedule. This pipelining property of the RDPS algorithm naturally facilitates the concurrent generation of task schedules and strikingly reduces the time to generate the desired β and task schedule. For example, for a given a dual core processor, if the current value of β is β_0 and current reliability does not meet the specified RDPS stop-condition, the next iterations of RDPS with $\beta = \beta_1$ and $\beta = \beta_2$ can be run simultaneously on core 1 and core 2, respectively, as is illustrated in Fig. 4.2. It is shown in Section 4.5 that the time to generate a desired task schedule

for a task set with up to 100 tasks is in the order of seconds.

4.4.2 Generate Task Schedules Using a LP Solver and Sequential Rounding Technique

Due to the difficulties to solve ILP program efficiently, linear program (LP) with sequential rounding is utilized to compute task schedules by relaxing the integer constraints of the ILP. LP with sequential rounding is a well-known technique and has been extensively investigated in the literature [67].

The integer constraint of the ILP definition given in Equation (4.11), that is, $A_{m,l,n}$ is either 0 or 1, is relaxed and it becomes a real number such that $0 \leq A_{m,l,n} \leq 1$. A commercial or open source linear program solver is used to solve the energy optimization problem defined in Equation (4.10) under the

relaxed constraint. The output of the LP solver, $A_{m,l,n} \in R$ for $m = 1, 2, \dots, M$, $l = 1, 2, \dots, L$, and $n = 1, 2, \dots, N$, indicates that each task can be assigned to more than one processors, and the task assigned to a specific processor can run at one or more frequency levels. This will incur extra overhead owing to task migration among processors and task frequency switching on the same processor. A well-known sequential rounding technique is therefore utilized in this section to assign a task to one and only one processor and to set the operating frequency of the task at one and only one frequency level.

Assume the current output of the LP solver to be $A_{m,l,n}$, that is, the task τ_m is assigned to processor n . Further, assume $(A_{m,l_1,n} + A_{m,l_2,n} + \dots + A_{m,l_b,n}) = 1$ for $1 < l_1, l_2, \dots, l_b < L$, where b is a constant integer and $1 < b < L$. The execution of

task τ_m is hence to be scheduled at b different frequency levels.

In other words, the task is not to be scheduled at any other frequency levels and $A_{m,l,n} = 0$ for $l \neq l_1, l_2, \dots$, and l_b . Define two thresholds δ_{low} and δ_{high} for $0 < \delta_{low} < \delta_{high} < 1$. There exists three rounding cases based on the defined thresholds during task schedule generation, as is detailed below.

† If $A_{m,l,n} > \delta_{high}$ for $l = l_1, l_2, \dots$, or l_b , for example, $A_{m,l_1,n} > \delta_{high}$, set $A_{m,l_1,n}$ to 1 and reset $A_{m,l,n}$ to 0 for $l \neq l_1$. That is, the whole task is to be executing at the frequency level l_1 .

† Similarly, if $A_{m,l,n} < \delta_{low}$ for $l = l_1, l_2, \dots$, or l_b , for example, $A_{m,l_2,n} < \delta_{low}$, reset $A_{m,l_2,n}$ to 0. That is, the task is not to be scheduled at the frequency level l_2 . For the given $A_{m,l_2,n} = 0$, re-compute the values of $A_{m,l,n}$ for $l \neq l_2$

by re-solving the linear program.

† If $A_{m,l,n}$ for $l = l_1, l_2, \dots$, and l_b fall in $[\delta_{low}, \delta_{high}]$, the values of δ_{low} and δ_{high} need to be adjusted for the schedule rounding to continue. The new values of δ_{low} and δ_{high} mainly depends upon time requirements to generate the desired task schedule.

At this stage, the RDPS algorithm gives an energy optimum task-to-processor assignment without considering the feasibility of the schedule under stochastic fault occurrences. The reliability of each task in the generated task schedule needs to be derived and verified to meet the given reliability targets under stochastic fault occurrences, which is described in the next section.

4.4.3 Derive Task Reliability Using the Monte Carlo Simulation and Latin-Hypercube Sampling

The reliability of a task in a given task set is evaluated under fault occurrences of Poisson probability distribution using Monte Carlo simulation. Monte Carlo simulation is a technique that relies on probability distribution and random numbers to evaluate the reliability. It is often used for analyzing a system with significant uncertainty in inputs, and how the uncertainty affects the performance or reliability of the system. Oftentimes, the reliability of a task schedule is obtained in three major steps. First, the reliability evaluation of the current task schedule is started by generating the average fault arrival rate at various frequency levels at which tasks are scheduled. That is, the

parameter λ_l of the Poisson probability distribution needs to be calculated using Equation (4.3). Note that the λ_l for task τ_m scheduled at the frequency f_m is fixed. Hence, the parameters of the Poisson probability distribution of task τ_m for $m = 1, 2, \dots, n$ are fixed.

In step 2, the number of fault occurrences during the execution of each task is then generated based on the probability distribution of the task, and the execution time of the task is updated to include fault recovery overhead. In step 3, the feasibility of the generated task schedule is verified. The step 2 and step 3 constitute one sample of the Monte Carlo simulation. Repeat step 2 and 3 to take more than 10,000 Monte Carlo samples, and the reliability of the current task schedule is derived as the ratio of the number of samples where

the schedule is feasible to the total number of Monte Carlo samples. If the current reliability is greater than and yet close enough to the target reliability, the resultant task schedule is the desired schedule and the execution of the RDPS algorithm exits. Otherwise, the RDPS algorithm jumps from step *C* to step *A* and continues its execution, as is described in Fig. 5.3.

Although the Monte Carlo simulation described above exhibits relative generality and insensitivity to the number of stochastic fault occurrences, it is expensive for accurate reliability estimation of a task schedule. Therefore, the Latin Hypercube sampling method is adopted in this work to improve the efficiency of reliability estimation for a task schedule by sampling fault occurrences more systematically.

The Latin Hypercube sampling (LH), first described by McKay in [68], has been utilized in uncertainty analysis to generate multivariate samples of statistical distributions. For a one-dimensional variable-sampling, it starts by estimating the uncertainty of a variable using a probability distribution, dividing the variable area into intervals with identical probability, and generating a sample value for the variable in each interval.

A two-dimensional variable-sampling is used to illustrate the idea of LH sampling. Given two random variables (x,y) that are to be simulated. 4 simulation samples of the variable can be obtained by randomly generating 4 pairs of (x,y) . As is shown in Fig.5.5(a), these samples are randomly distributed in the simulation domain, and hence are not representative of the

two random variable.

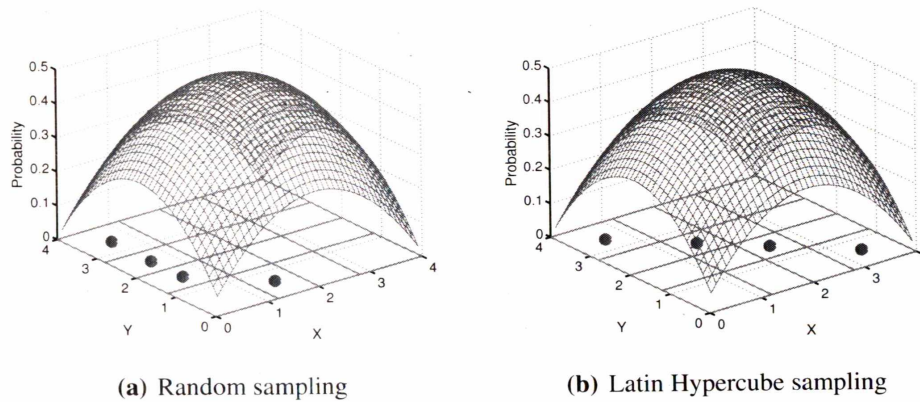


Figure 4.3: Random sampling versus Latin Hypercube sampling

LH sampling technique can be utilized to tackle the non-representative sampling issue. In LH sampling, the range of each variable is divided into equally probable intervals, each of which can associate with one and only one simulation sample. As is illustrated in Fig.5.5(b), the simulation domain is first divided into 4×4 grids of equal probability. When one grid is selected to generate a simulation sample, grids with the same row or column can not be the candidate grids for future sample

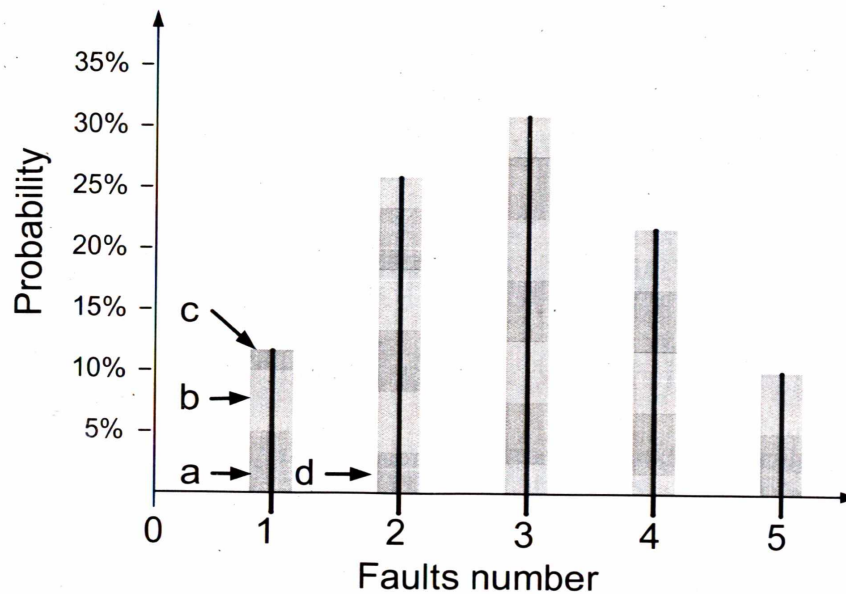


Figure 4.4: Adapt LH to discrete Poisson distribution

generation. As a result, representative simulation samples are generated, and the total number of simulation iterations can be significantly reduced.

LH technique is in general used for sampling random variables of continuous probability distribution. In this research, LH is adapted to sample random variables with discrete Poisson

probability distribution, which is visualized and illustrated in Fig. 4.4. Assuming the number of the required samples to be 20, the Poisson distribution is then divided into 20 equally probable intervals, each of which accounts for the probability of 0.05 in fault occurrences. Starting from the left of the figure, the probabilities of interval a , b , c , and d are 0.05, 0.05, 0.01, and 0.04, respectively. Since the probability of interval c is less than 0.05, the interval d of probability 0.04 is delimited such that the sum of probabilities of interval c and d is 0.05. One sample of fault occurrences is to be taken in each interval of probability 0.05. As a result, the numbers of fault occurrences associated with interval a and b are both 1, and the number of fault occurrences associated with interval c/d is to be either 1/0 or 0/2, depending upon the relative probabilities

of interval c and d . Similarly, the samples of fault occurrences for the remaining intervals can be derived. 200 samples of fault occurrences are taken in the proposed scheme for feasibility probability evaluation.

Table 4.1

The average energy consumption (in mJ) of task sets with varying sizes and the average CPU time (in seconds) of the proposed RDPS algorithm ($0 < \beta < 1$) for target reliability of 0.99.

Task set size	1-core					2-cores					4-cores				
	$\beta = 0$	$0 < \beta < 1$ (RDPS)			$\beta = 1$	$\beta = 0$	$0 < \beta < 1$ (RDPS)			$\beta = 1$	$\beta = 0$	$0 < \beta < 1$ (RDPS)			$\beta = 1$
	E_B	E_R	CPU_4	E_{WR}	E_W	E_B	E_R	CPU_4	E_{WR}	E_W	E_B	E_R	CPU_4	E_{WR}	E_W
10-40	158	187	29.1	19.1%	231	152	185	32.2	17.1%	223	165	192	87.1	21.6%	245
41-70	273	298	36.2	35.4%	461	268	292	49.3	37.9%	470	271	316	102.7	31.7%	463
71-100	521	590	45.7	20.1%	738	513	577	71.2	24.0%	759	542	595	172.3	20.9%	752

Table 4.2

Comparison of the proposed stochastic RDPS algorithm ($0 < \beta < 1$) with the best case ($\beta = 0$) and worst case approach ($\beta = 1$) in energy consumption (in mJ). CPU time is in seconds.

# of tasks→ (# of cores)	$\beta = 0$ (Fault free)			$\beta = 1$ (Worst case faults)			$0 < \beta < 1$ (Proposed RDPS: Stochastic fault occurrences)					
	E_B	Reliability	CPU_1	E_W	Reliability	CPU_1	E_R	Reliability	CPU_1	CPU_4	E_{WR}	CPU_1/CPU_4
31→(1)	185	0.0	0.7	270	1.0	0.8	201	0.99	69.4	22.1	25.6%	3.1 ×
43→(1)	277	0.0	0.7	466	1.0	0.7	302	0.99	66.9	23.2	35.2%	2.9 ×
78→(1)	445	0.0	0.9	652	1.0	1.0	481	0.99	39.0	23.5	26.2%	1.7 ×
56→(2)	341	0.0	1.2	498	1.0	1.2	373	0.99	65.1	35.1	25.1%	1.9 ×
88→(2)	582	0.0	1.4	841	1.0	1.5	639	0.99	72.9	54.3	24.0%	1.3 ×
25→(4)	144	0.0	1.1	210	1.0	1.1	172	0.99	73.2	42.4	18.1%	1.7 ×

4.4.4 Complexity Analysis

The proposed RDPS algorithm consists of three iterative steps. Of the three steps, the step *A* and the step *C* take constant time. Hence, the computation overhead of the proposed RDPS algorithm mainly depends on the overhead to derive the task schedule using ILP solver in step *B*.

In step *B* an interior point technique-based ILP solver is adopted. The complexity of the interior point ILP solver is $O(z^3)$, where z is the total number of variables in the ILP formulation [69]. The variables involved in the proposed formulation are $A_{m,l,n}$, S_m , $S_{min(i,j)}$, $S_{max(i,j)}$, $b_{i,j}$, $g_{i,j}$, and $h_{i,j}$. The number of the $A_{m,l,n}$ and S_m is mnl and m , respectively, and the total number of all other variables is $5m^2$. Thus, z could

be written as $m(5m + nl + 1)$. After each iteration of the ILP solver, the sequential rounding technique is utilized to round the variable of $A_{m,l,n}$. It takes one ILP solver iteration to round one variable and it takes mnl iterations of the ILP solver to round all the variables of the $A_{m,l,n}$ in the worst case. After rounding, the step B of the RDPS algorithm is completed, and its computation overhead is $O(mnlz^3)$.

Since the step A and C of the RDPS algorithm take constant time, the computation overhead of the RDPS algorithm can be estimated by the computation overhead of the step B times the iteration number of the RDPS algorithm. The number of iterations of the RDPS algorithm is determined by how the β value is picked in an iteration. Since a binary search-based approach is utilized in the proposed RDPS algorithm to select

the next value of the β , and the β -based RDPS algorithm is able to compute the desired task schedules in parallel, the number of iterations of the RDPS is deemed to be the logarithm of the number of the samples of the β . Oftentimes, the samples of the β is limited. For 16 samples of the β , the computation overhead of the RDPS algorithm is $\log_2^{16} \times O(mnlz^3) = O(4mnlz^3)$. This computation overhead is acceptable for RDPS because it is an offline algorithm.

4.5 Numerical Results

Extensive experiments were carried out over a simulated multiprocessor system to validate the proposed schemes for energy efficiency and running time. It is assumed that the multiprocessor supports 4 discrete voltage levels, which are

0.5V, 0.65V, 0.8V, and 1.0V. The values of the dynamic power and the leakage power of the processor, scaled to 70nm technology based on the technology scaling trend, is adopted from [65]. The proposed reliability-driven parallel task scheduling algorithm RDPS was implemented in C++, and the simulation was performed on a machine with Intel Core 2 Quad 2.4GHz processor and 8GB memory. Experimental results were averaged 1000 tasksets.

Experimental results were performed over various task sets. Task sets were randomly generated by assuming varying deadlines. Task execution times are in the range of 10-40ms. Transient fault occurrences are assumed to follow the Poisson probability distribution, and the average fault arrival rate at the lowest processor speed is assumed to be 4.

Three designing approaches, that is, the best case, the worst case, and the proposed stochastic RDPS approach, are compared in energy savings and computational complexity under a given task level target reliability. Let E_B , E_W , and E_R denote the energy consumption of a task set under the best case fault occurrences ($\beta = 0$), the worst case fault occurrences ($\beta = 1$), and the stochastic fault occurrences ($0 < \beta < 1$), respectively. E_R in fact indicate energy consumptions of the proposed RDPS algorithm. Then let $E_{WR} = \frac{(E_W - E_R)}{E_W} \times 100\%$ denote energy savings of the proposed RDPS scheme ($0 < \beta < 1$) when compared to the approach of designing a system under the worst case of fault occurrences ($\beta = 1$). The proposed RDPS algorithm can efficiently compute the desired task schedule in parallel. Several notations have been

introduced to indicate the speedup capability of the proposed scheme. Let CPU_1 be the CPU time it takes to compute the desired task schedule using one core of the Core 2 Quad processor, and CPU_4 be the CPU time computing the desired task schedule using all 4 cores of the Core 2 Quad processor, then CPU_1/CPU_4 indicates the speedup of the proposed parallel RDPS algorithm comparing to the technique by using a single core.

Table 4.1 shows the average energy consumption of task sets with varying sizes and the average CPU time of the proposed RDPS algorithm ($0 < \beta < 1$) for a common target reliability of 0.99. Tasks are assigned to a 1-core, 2-core, and 4-core system respectively for both $\beta = 0$ and $\beta = 1$. For the proposed RDPS algorithm ($0 < \beta < 1$), tasks are always assigned to all 4 cores

of the Core 2 Quad processor due to the scheduling parallelism property of the scheme. The proposed parallel RDPS algorithm ($0 < \beta < 1$) saves energy up to 37% by comparing to the approach of designing for the worst case faults ($\beta = 1$). For instance, when tasks of a given task set the size of which is 41-70 are assigned to a 2-core system, the RDPS scheme ($0 < \beta < 1$) consumes 37.9% less energy when compared to the designing approach for the worst case faults ($\beta = 1$). The CPU time of the RDPS algorithm for task sets with 10 to 100 tasks is in the order of seconds. For example, the average CPU time for the task set with 41-70 tasks assigned to a 2-core system is 49.3 seconds.

Table 4.2 presents the comparison of the designs under the best case fault occurrences ($\beta = 0$) and the worst case fault

occurrences ($\beta = 1$) with the proposed design under the stochastic fault occurrences ($0 < \beta < 1$) in energy consumption and CPU time. The given target reliability is assumed to be 0.99. The speedup performance of the proposed parallel RDPS algorithm ($0 < \beta < 1$) is also listed in the table. Let $\# \text{ of tasks} \rightarrow (\# \text{ of cores})$ denote the number of tasks and the number of cores where the tasks are to be assigned. For example, $31 \rightarrow (4)$ signifies the scheduling of 31 tasks to a 4-core system. Recall that the RDPS paralleling is enabled by the fault adaptation variable β , hence the fault free scheduling where $\beta = 0$ and worst case scheduling where $\beta = 1$ can not be parallelized. This is because the β is constant in the later two cases.

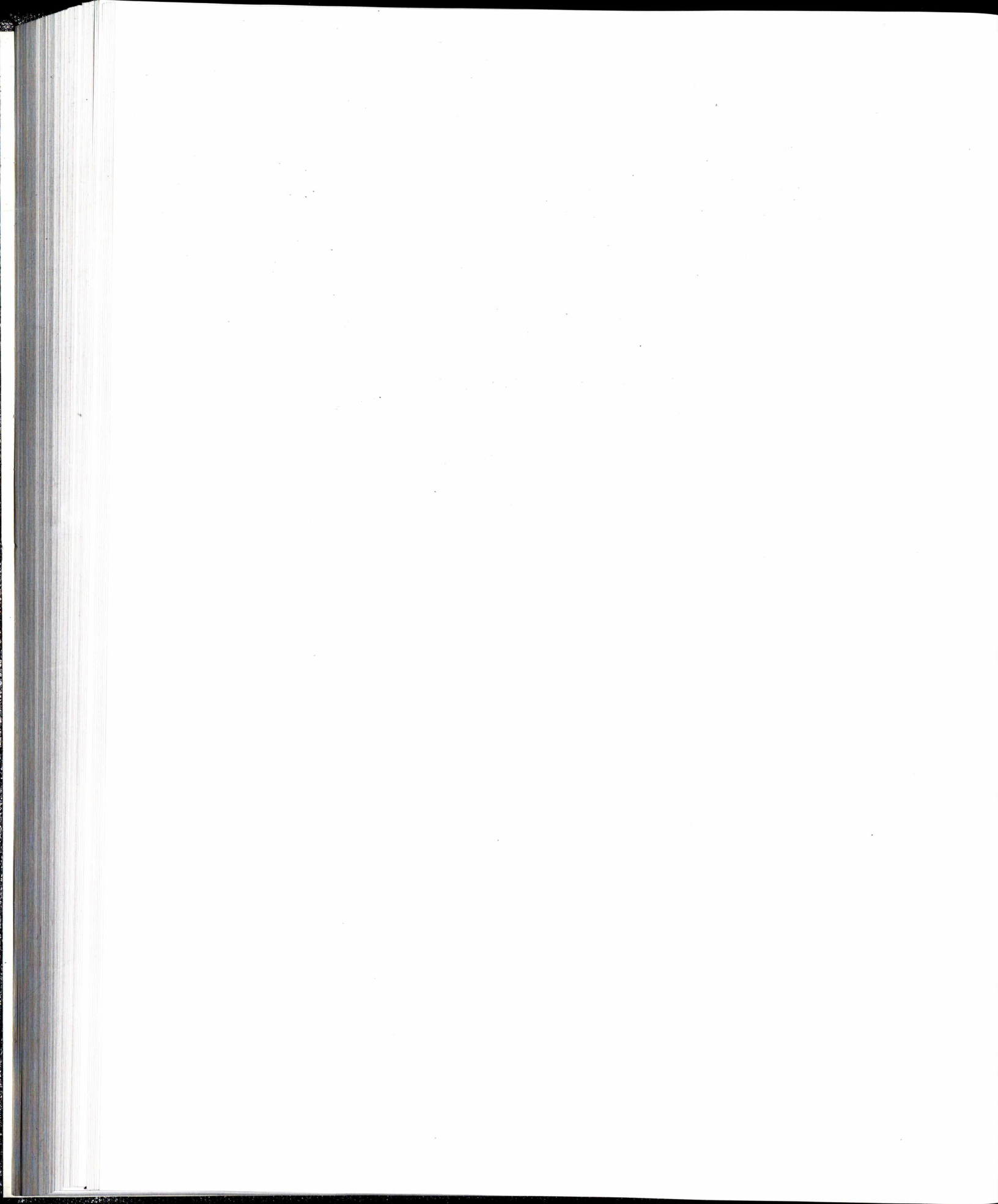
As is shown in Table 4.2, on one hand, the schedule reliability

is 0 when $\beta = 0$. This is primarily due to the fact that tasks in given task sets are scheduled under fault-free conditions while the schedule evaluation is performed under stochastic fault occurrences. On the other hand, the reliability of a task schedule are 1.0 when $\beta = 1$, indicating an over-designed energy inefficient system. For a given target reliability of 0.99, the proposed parallel RDPS algorithm ($0 < \beta < 1$) consumes significantly less energy when compared to the designing approach under the corner case faults ($\beta = 1$) and achieves a speedup of up to $3.1\times$. For example, for $31 \rightarrow (1)$, the energy savings of the proposed parallel RDPS algorithm ($0 < \beta < 1$) with regard to the corner case designing approach ($\beta = 1$) is 25.6%, and the speedup of the proposed parallel RDPS algorithm ($0 < \beta < 1$) is $3.1\times$.

4.6 Conclusion

This project presents a novel reliability-driven parallel task scheduling scheme that is featured by a fault adaptation variable β . The proposed RDPS scheduling scheme considers both dynamic power and leakage power, is able to handle the scheduling of both independent tasks and dependent tasks, and is capable of scheduling tasks with varying deadlines. The RDPS algorithm optimizes system energy consumption under stochastic fault occurrences and accelerates the computing of the desired task schedule by utilizing techniques such as the β -enabled scheduling parallelism, sequential rounding, and Latin Hypercube sampling-based Monte Carlo simulation. Experimental results have shown that the proposed parallel

RDPS scheme saves energy more than 15% by comparing to the approach of designing for the corner case of fault occurrences, and exhibits a speedup of up to $3.1\times$ on a machine with Intel Core 2 Quad processor.



Chapter 5

Smart Home Uncertainty-Aware Household Appliance Scheduling¹

5.1 Introduction

The power grid has significant amount of uncertainties due to its complex capacity. The integration of advanced technologies such as renewable energy generation including wind farms

¹The material contained in this chapter has been accepted by IEEE Transactions on Smart Grid for publication.

and solar cells introduces further complexity and challenge to various controllers at all levels of the power grid [70]. Salient communication and information technologies have been explored in utility industry to handle the increasing complexity [71].

In a smart grid infrastructure which targets to facilitate the modernization of the classical power grid, utility companies explore demand side management (DSM) technology [72, 73, 74, 75, 76]. It enables the integration of various renewable energy resources which include solar, wind and bio energy [75, 76]. Demand side management technology can help shift the energy consuming workload from peak time to off-peak time for the purposes such as load balancing and monetary expense reduction [77], which is critical in a smart home

system.

There are multiple components in a smart home system such as various household appliances, energy storage and renewable generation component such as photovoltaic (PV) arrays. Typical examples of household appliances include plug-in hybrid electric vehicles (PHEVs), air conditioners, refrigerators, and laundry machines. Figure 5.1 depicts the average household electricity in California for a 24 hours time period [78] with consideration of PHEVs. In this household appliances profile, the PHEV has the most significant portion of electricity consumption among all home appliances. The air conditioning, refrigerator, lighting and laundry ranks from the second to fifth, respectively. It is obvious that as a new member of home appliances, the PHEV plays a very important

role, since the electricity consumption of the PHEV accounts for 37% of the total household electricity. As the result, the PHEV has obtained considerable research attentions recently. In [79], the authors propose a demand response strategy which can help to alleviate load peaks. In [80], the authors propose a scheme to determine which household appliance needs to be controlled considering the increasing penetration level of PHEVs. Formulation of the home appliances models for traditional home appliances was investigated in [81] [82].

Energy storage such as high capacity batteries are often used to store energy generated from photovoltaic arrays. Each customer installs a smart meter [83, 74, 84]. There is a scheduling unit which implements the workload shifting mentioned above in the smart meter. It periodically receives

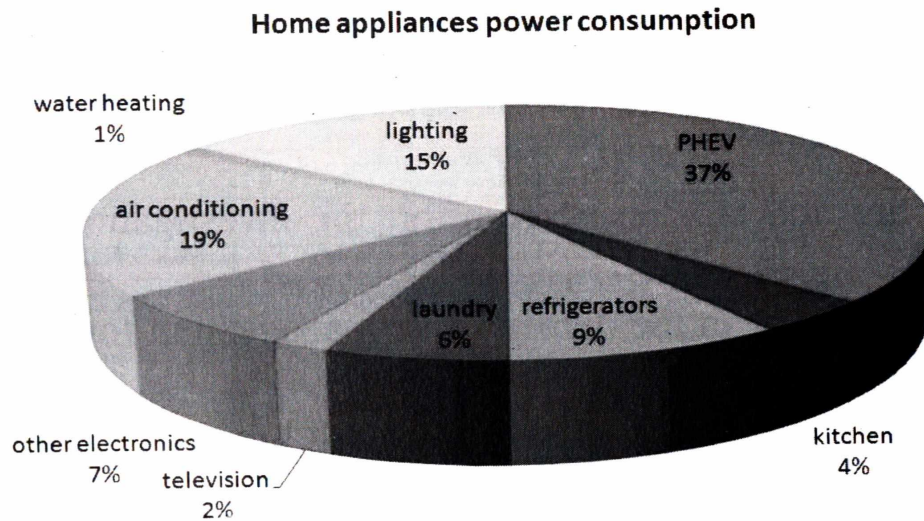


Figure 5.1: Home appliances electricity consumption in average

the updated pricing information from the utility companies, and its scheduling unit arranges different household appliances for operation during different time periods. It is effective in reducing the monetary expense charged to end users since different electricity rates can be applied at different time periods in the popular real-time pricing model [85, 86, 87, 88]. Refer to Figure 5.2 for a smart home scenario. This paper aims to minimize the monetary expense of a single customer

through optimally scheduling by considering operating speed, appliance launch time under the real-time pricing environment.

There are lots of research effort in investigation of the scheduling issue in the demand side management for reducing the monetary expense of the customer and peak-to-average ratio in load demand. Kim et al. [89] investigated the energy consumption scheduling problem with time-varying prices known in advance to customers. Optimal scheduling algorithms that bring significant gains to customers were derived to find a series of price thresholds by using stochastic dynamic programming.

Currently, there are various dynamic pricing rates, and table 5.1 concludes the most popular rates.

Table 5.1
Dynamic pricing model comparison [3].

Pricing Model	Description	Current state
Time of Use (TOU)	Charge more during on-peak hours Charge less during off-peak hours	Most common time-based rate
Super peak TOU	Similar to the TOU But has narrow peak window	
Critical Peak Pricing (CPP)	Charge more during on-peak hours with a certain number of days	Commonly tested as pricing pilots
TOU and CPP Combination	Charge more during on-peak hours higher price with a certain number of days	
Peak Time Rebate (PTR)	Flat rate with a rebate of reducing household load during the peak hours	Used for commercial and industry customers
Flat Real Time Pricing (RTP)	Flat rate with hourly variation across all hours of the year	Tends to be changed

There are many existing works based on these rates. In [90], authors propose variational inequality models for electricity markets with time-of-use (TOU) pricing. In [91] a residential side scheduling technique with a real-time pricing model is proposed. In [90], authors use analyze the price distribution and demand distribution for the U.K. market. In [92, 93], the authors presented an energy consumption scheduling heuristic to reduce the peak load in individual homes or buildings with reasonable computation time. The operations of household appliances are classified into preemptive and non-preemptive

operations, and the scheduling for preemptive operations is based on the schedule of the non-preemptive operations. In [94], the authors proposed a power scheduling protocol for demand response in smart grid systems. A joint media access and appliance scheduling approach was developed to manage the power usage of appliances so that total power demand is kept below a target value. Extensive research on the demand side load management also has been performed for a neighborhood with multiple customers. In [73, 74], the authors considered custom appliances scheduling by using smart meter in the power grid, based on game theory. The technique from this work can be used to schedule home appliances electrical energy consumption for the customer in the neighborhood. The presented algorithm aims to minimize the total electrical costs,

electricity charges of individual customers and peak-to-average ratio. In [95], authors proposed a technique which considers dynamic pricing to achieve an accumulated house appliance load. By sharing the, distributed scheduling algorithms were designed to reduce total electrical energy consumption as well as the peak-to-average ratio, and improve the overall load profile of the system. In [96], the authors proposed a methodology to manage the energy production, consumption, and storage.

In the above recent research works which focus on reducing the monetary expense of customers, the stochastic characteristics of customer energy consumption patterns are not considered, which is however quite important. Stochastic design technique itself has been investigated in the literature in varying contexts

such as stochastic security constrained unit commitment (SCUC) design [97] and wireless base station construction [98]. In [97] and [98], the uncertain problem is directly approached using the stochastic programming technique, and the benders decomposition method is utilized to reduce the time to obtain the optimized results. In [99, 100, 101], the authors presented stochastic models that minimize the total cost of generators and transmission system with the consideration of the power system uncertainties, and inaccuracies in load forecasting. Monte Carlo simulation is utilized to model the uncertainties, and stochastic problem is also used for subproblems. However, no related work on stochastic optimization has been found in the area of smart home design.

In addition, the energy storage and renewable generation such

as wind and solar are not considered in most of the related research works. With emerging requirements for renewable portfolio standards, wind and solar generation become a must-take resources in many countries of the world and about 30 of 50 U.S. states [102, 75]. For instance, the State of California requires 33% of the total energy generated from renewable resources by 2020. However, the predicted energy generation can be significant different than the actual energy generation from renewable resources. For example, the solar energy generated from a PV panel may vary with the changes in sun irradiation level, the angle of the sun, or even the lasting time of cloud shadow. This intermittent nature of renewable resources imposes significant challenge in designing salient scheduling techniques considering renewable generation [75].

Furthermore, most existing smart home energy scheduling works do not consider variable frequency drive (VFD) which is however a very important technique for expense minimization and load balancing. Basically, the task execution time is determined by assuming task being scheduled to operate at a typical frequency level. The execution time varies with different frequencies. For example, if a micro-wave oven originally needs 10 minutes to cook a dish when operating at its nominal frequency 1.5GHz, it might need only 5 minutes when operating at 3GHz. As far as the performance is concerned, it is certainly desirable to schedule the household appliance to run at the high frequency.

Normally, each household has a certain limitation on the total appliance load. When the total load demand of household

appliances exceeds the given load limit of the household, the home power network trips out. This will lead to degradation of customer comfortableness. *The probability that the home power network trips out during a time interval is defined to be the trip rate.* Or course, the scheduler should try to avoid tripping out. However, since there are uncertainties in the energy consumption of household appliances as well as renewable generation, one can only minimize the trip rate (to a very small value) in practice. Thus, it is desired for customers to set a trip rate constraint (e.g., 0.5%) such that the trip rate of the scheduling solution is no greater than the constraint.

In this paper, a home appliances scheduling algorithm is proposed which can be used to minimize the monetary expense without compromising the comfortableness of customers.

Precisely, the proposed operation scheduling algorithm takes as inputs the time-varying pricing information released by power utility companies ahead of time, distributed renewable generations and energy storage, and the customer-defined target trip rate. It generates an operation schedule over a pre-defined time domain (called horizon) that minimizes the customer monetary expense and meet the customer-defined trip rate. Our algorithm features the consideration of the uncertainties in household appliance operation time and energy consumption and energy generated from the renewable resources. To handle the uncertainty in household appliances, a stochastic scheduling algorithm which involves an energy adaptation variable β to model the uncertainty in energy consumed by an appliance is designed. On the other hand, since the

real-time energy generated from renewable energy resources is much more difficult to predict precisely, its uncertainty will be handled using online scheduling algorithm, which adapts the offline schedules to the online schedules of renewable energy. The proposed algorithm can also handle the scheduling of operations of VFD-equipped appliances. The simulation results demonstrates that when compared to a worst case design where an appliance is assumed to consume the maximum amount of energy, the proposed design that considers the stochastic energy consumption patterns achieves up to 24% monetary expense reduction without violating the target trip rate of 0.5%. When compared to a natural greedy algorithm which models typical household appliance operations in the traditional home scenario, the proposed deterministic linear

programming based scheduling scheme achieves up to 53% monetary expense reduction. Furthermore, the proposed energy consumption scheduling algorithm can always generate the scheduling solution within 10 seconds, which is fast enough for household appliance applications.

The project is organized as: Section 5.2 describes the system architecture and models, and defines the optimization problem. Section 5.3 formulates the scheduling problem into a linear programming problem and proposes the offline operation scheduling algorithm that minimize the customer monetary expense. In Section 5.3.3, the offline operation schedule is adapted to the intermittent behavior of the renewable generation. Section 5.4 presents the simulation results, and Section 5.5 concludes the paper.

5.2 System Architecture and Models

This paper aims to minimize the monetary expense of the customer through optimally scheduling the operating and execution time window of each appliance considering uncertainties under the real-time pricing environments. The following subsections describe the system models and problem definition.

5.2.1 Residential Customer Model

As is illustrated in Figure 5.2, a residential unit may include various household appliances such as air conditioners, space heaters, washers, refrigerators, plug-in hybrid vehicles, etc. Let a denote an appliance and A be set of a . For each $a \in A$, there is

\mathbf{X} used to denote the vector of electrical energy consumption, which can be defined as

$$\mathbf{X} \triangleq [x_a^1, \dots, x_a^T],$$

where T denotes total number of time units before the scheduling decision on energy consumption is to be made. For a pricing structure that releases price information one day ahead [103], the scheduler arranges the operation of appliances for the next 24 hours. The scheduling horizon is 24 hours in this scenario. The resolution of scheduling horizon can be hours, minutes, or even seconds, depending on the available pricing information and the computing capability of the scheduler in the smart meter. For example, since the Ameren Illinois Power Corporation releases hourly price information one day ahead

[103], the resolution of scheduling can be set to hourly in this case.

For each time unit $\tau \in \mathbf{T} \triangleq [1, 2, \dots, T]$ in the horizon of scheduling, the element x_a^τ of the vector \mathbf{X} denotes the energy consumption of a during the interval τ . Suppose that there are two user-defined time instants δ_a and θ_a which indicate the start time and end time of the operation of the appliance a , respectively. It is clear that $\delta_a < \theta_a$ holds and the appliance a consumes no energy beyond the interval of $[\delta_a, \theta_a]$. In other words, $x_a^\tau = 0$ for $\tau < \delta_a$ or $\tau > \theta_a$.

It is assumed that each appliance $a \in A$ has a maximum energy level in the interval of τ , which is defined to be the rated power and denoted by P_a . For example, let a denote a laundry machine

and it may operate at the power of up to $P_a = 5.6$ kW per hour. It is clear that $x_a^\tau \leq P_a$ holds, where x_a^τ is the actual energy consumed by the appliance a during the interval of τ . Oftentimes, there is a limit on the total energy consumed by various appliances of a household in the time interval of τ . Let L_A^τ denote the value of the energy limit, then the inequality

$$\sum_{a \in A} x_a^\tau \leq L_A^\tau \quad (5.1)$$

holds for $\tau \in \mathbf{T}$. When the above constraint is violated, the home power network will be tripped out.

The total energy consumed by an appliance $a \in A$ during the scheduling horizon T is given by $\sum_{\tau \in \mathbf{T}} x_a^\tau$ and denoted by E_a^T . E_a^T is essentially a random variable. For instance, the laundry time of an advanced laundry machine depends on the load,

which is usually a Gaussian distribution. As a result, the energy consumed by the machine follows the Gaussian distribution of the probability.

Let μ denote the mean and σ denote the standard deviation of the random variable, and let γ_a^{min} and γ_a^{max} denote the minimum and maximum value of the random variable, respectively. For a random variable following a Gaussian distribution, with more than 99% confidence its maximum deviation from the mean is bounded by 3σ . Thus, we set $\gamma_a^{min} = \mu - 3\sigma$ and $\gamma_a^{max} = \mu + 3\sigma$.

During our optimization, since it is difficult to directly solve a mathematical program with uncertainty the energy optimization problem is transformed into a set of deterministic optimization problems without random variables (refer to

Section 5.3). In this work, motivated from [104], a variable β , referred to as energy adaptation variable, is introduced to model the uncertainty in energy consumed by an appliance. The actual energy consumed by appliance $a \in A$ in the scheduling horizon T can be expressed as a function of β , γ_a^{min} , and γ_a^{max} , that is,

$$E_a^T = (1 - \beta) \times \gamma_a^{min} + \beta \times \gamma_a^{max}, \quad (5.2)$$

where $0 \leq \beta \leq 1$. The energy consumption of appliance a during T is γ_a^{max} if $\beta = 1$ and is γ_a^{min} if $\beta = 0$.

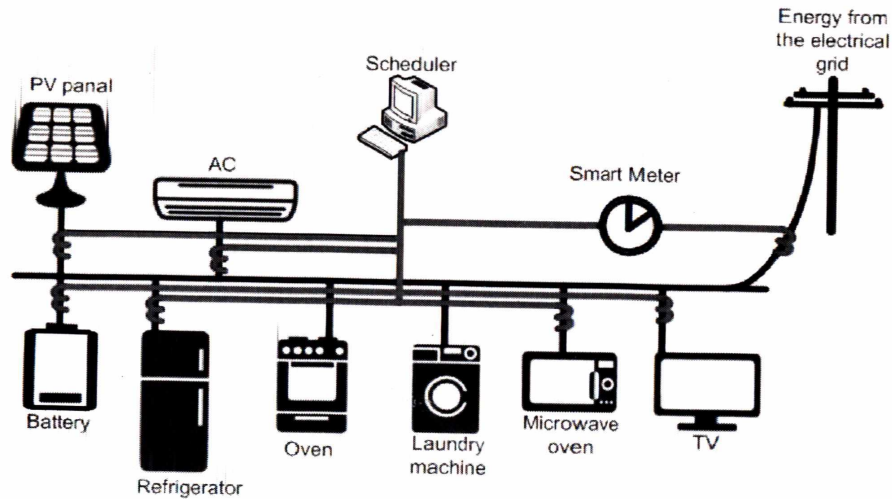


Figure 5.2: A smart home scenario

5.2.2 Pricing Model

In the literature, a lot of time related pricing models have been proposed, such as day-ahead pricing (DAP), inclining block rates (IBR), time-of-use pricing (TOU), critical-peak pricing (CPP), etc. [91]. Among them, the RTP and IBR models have been extensively investigated from various perspectives [105, 106, 107, 108, 109, 110, 111, 112]. Basically, in RTP pricing model prices could be different for different time intervals while they are flat within each time interval. In contrast, in IBR pricing model prices remain the same over time while incurring an increase when the energy consumption of a residential unit reaches a pre-determined threshold. Combining RTP and IBR pricing models, which can reflect both the

fluctuating wholesale price and the energy consumption level, would lead to the promising replacement model of the current flat rate tariffs [91].

As is in [91], let $c^\tau(E_A^\tau)$ denote the price of the energy consumed by all household appliances in the interval of τ , then $c^\tau(E_A^\tau)$ can be formulated as

$$c^\tau(E_A^\tau) = \begin{cases} b_1^\tau & 0 \leq E_A^\tau \leq l_{th}^\tau \\ b_2^\tau & E_A^\tau > l_{th}^\tau \end{cases} \quad (5.3)$$

where $b_1^\tau, b_2^\tau > 0$ are differentiated price and l_{th}^τ is the energy consumption threshold in the time interval of τ .

It is assumed that real-time pricing parameters b_1^τ, b_2^τ , and l_{th}^τ are known for users ahead of time. For example, one day ahead pricing data released by Ameren Illinois Power Corporation

[103] are available online. These data can be utilized to schedule the operation of appliances for monetary expense optimization.

Note that the proposed energy consumption scheduling scheme is independent of pricing models or pricing prediction models as is described in [113, 114]. Thus, the proposed scheme can be combined with other pricing models to minimize the customer monetary expense.

5.2.3 Photovoltaic (PV) Model

With the increasing the penetration of grid-connected photovoltaic systems, extensive research has been conducted on maximizing output power from an PV panel. For example,

various Maximum Power Point Tracking (MPPT) techniques are summarized and compared in [115]. It is shown that a two-stage [116, 117, 118] MPPT technique can be used for residential areas, which can be further combined with the irradiance forecasting scheme presented in [119] to estimate the energy output. Taking the PV array output energy as input, the proposed energy consumption algorithm attempts to maximize the benefit from solar energy so as to minimize the overall monetary expense of the residential customer.

Note that the proposed scheme is also independent of specific solar power prediction models. In other words, the proposed scheme can be jointly utilized with any other solar power prediction approach. For the illustration purpose, in this work the power generated from a photovoltaic system is assumed

to follow the probabilistic distribution function derived from historical data.

5.3 The Proposed Stochastic Scheduling Algorithm For Household Appliances

Our algorithm consists of three parts. The first part is a linear programming based deterministic scheduling algorithm. The second part is a stochastic scheduling technique based on the deterministic linear programming scheduling technique to handle the uncertainty in energy consumption and runtime of household appliances. The last part is the online runtime scheduling, which can effectively handle the uncertainty in the energy generation from the photovoltaic system.

various Maximum Power Point Tracking (MPPT) techniques are summarized and compared in [115]. It is shown that a two-stage [116, 117, 118] MPPT technique can be used for residential areas, which can be further combined with the irradiance forecasting scheme presented in [119] to estimate the energy output. Taking the PV array output energy as input, the proposed energy consumption algorithm attempts to maximize the benefit from solar energy so as to minimize the overall monetary expense of the residential customer.

Note that the proposed scheme is also independent of specific solar power prediction models. In other words, the proposed scheme can be jointly utilized with any other solar power prediction approach. For the illustration purpose, in this work the power generated from a photovoltaic system is assumed

to follow the probabilistic distribution function derived from historical data.

5.3 The Proposed Stochastic Scheduling Algorithm For Household Appliances

Our algorithm consists of three parts. The first part is a linear programming based deterministic scheduling algorithm. The second part is a stochastic scheduling technique based on the deterministic linear programming scheduling technique to handle the uncertainty in energy consumption and runtime of household appliances. The last part is the online runtime scheduling, which can effectively handle the uncertainty in the energy generation from the photovoltaic system.

5.3.1 Linear Programming based Deterministic Scheduling

A residential unit consumes solar energy in addition to the energy from electrical grid. The solar energy from PV could be consumed by a residential unit, stored in a battery, or wasted if the battery is full. It is assumed that the price of energy from electrical grid is higher than that of the per-unit cost of solar operation and maintenance, and customers prefer to use solar energy.

Let y_u^τ and e_s^τ denote the energy from electrical grid and the energy produced by the PV system in the time interval of τ , respectively, and let c_u^τ and c_s^τ denote the unit price of the energy from electrical grid and solar energy, respectively, in the time interval of τ , then the objective function is given by Equation

(5.4), where T is the scheduling horizon. b_c denotes the cost of battery and I_c denotes the one-time installation cost of solar panel, both normalized to the scheduling horizon T . Note that the unit price of the solar energy (c_s^τ) essentially indicates the per-unit cost of solar operation and maintenance.

The energy consumed by all appliances in the interval of τ is typically upper bounded by a constant value L_A^τ , which is explained in Equation (5.1) and rewritten in Equation (5.5). If the total energy consumption in a residential unit exceeds the limit L_A^τ , then the electricity supply to the unit trips out.

It is assumed that the energy consumed by an appliance $a \in A$ during the scheduling horizon of T follows a distribution of probability. The mean of the energy consumption is denoted

by E_a^T . In the offline scheduling, this mean value is taken as the energy consumed by the appliance in the time span of T , as is given in Equation (5.6). Equation (5.7) shows that the energy consumed by an appliance $a \in A$ in the interval of τ , which is denoted by x_a^τ , is less than or equal to the rated power of the appliance, which is denoted by P_a . Equation (5.8) indicates that an appliance does not consume any energy beyond the interval defined by its operation start time δ_a and operation end time θ_a .

Let y_s^τ and y_b^τ denote the solar energy and the battery energy used in the time interval of τ , respectively. Then the energy consumption of household appliances in the interval τ is the sum of the energy from the grid, the solar energy directly from the PV system, and the battery energy, which is given in Equation (5.9), where y_u^τ is the energy from the grid.

As is described in Equation (5.10), part of the solar energy, which is denoted by y_s^τ , is provided to household appliances, and the remaining solar energy, which is denoted by z_s^τ , is stored in the battery. Since there could be waste of solar energy, the solar energy e_s^τ produced by the PV system is greater than or equal to the sum of the consumed and stored solar energy.

Equation (5.11) describes the energy constraint on the battery.

Let z_b^τ denote the energy left of a battery at the beginning of the time interval τ , and $z_b^{\tau-1}$ denote the energy left of a battery at the beginning of the previous time interval $(\tau - 1)$.

Similarly, the $z_s^{\tau-1}$ and $y_b^{\tau-1}$ denote the solar energy stored in the battery and the battery energy consumption of all tasks in the immediate previous interval of $(\tau - 1)$, respectively. Equation (5.11) indicates that the current battery energy in the time

interval of τ (z_b^τ) equals the remaining battery energy ($z_b^{\tau-1}$) plus the solar energy stored in the battery ($z_s^{\tau-1}$), and minus the battery energy provided for appliances in the immediate previous time interval ($y_b^{\tau-1}$).

In general, the lifetime and price of a battery depends on the total energy throughput of the battery, which is a fixed value by ignoring other aging effects. As a result, the price of a battery can be normalized to the scheduling interval of τ . Let b_u^τ denote the price of the battery with respect to the interval of τ and b_c denote the price of the battery with respect to the scheduling horizon T . The b_c is then given by Equation (5.12), where z_s^τ is the solar energy charged to the battery in the interval of τ .

Considering the grid electricity cost ($y_u^\tau \times c_u^\tau$), the solar

operation and maintenance cost ($e_s^{\tau} \times c_s^{\tau}$), the battery cost (b_c), and the one-time installation cost of solar panel (I_c), the optimization for the customer monetary expense can be formulated as follows.

$$\text{minimize: } \sum_{\tau \in \mathbf{T}} (y_u^\tau \times c_u^\tau + e_s^\tau \times c_s^\tau) + b_c + I_c \quad (5.4)$$

$$\text{subject to: } \sum_{a \in A} x_a^\tau \leq L_A^\tau, \forall \tau \in \mathbf{T} \quad (5.5)$$

$$\sum_{\tau \in \mathbf{T}} x_a^\tau = E_a^T, \forall a \in A \quad (5.6)$$

$$x_a^\tau \leq P_a, \forall a \in A, \tau \in \mathbf{T} \quad (5.7)$$

$$x_a^\tau = 0, \forall a \in A, \tau \notin [\delta_a, \theta_a] \quad (5.8)$$

$$\sum_{a \in A} x_a^\tau = y_b^\tau + y_s^\tau + y_u^\tau, \forall \tau \in \mathbf{T} \quad (5.9)$$

$$y_s^\tau + z_s^\tau \leq e_s^\tau, \forall \tau \in \mathbf{T} \quad (5.10)$$

$$z_b^\tau = z_b^{\tau-1} + z_s^{\tau-1} - y_b^{\tau-1}, \tau \in [2, \dots, T] \quad (5.11)$$

$$b_c = \sum_{\tau \in \mathbf{T}} z_s^\tau \times b_u^\tau \quad (5.12)$$

Variable Frequency Drive (VFD) technology has been widely adopted in household appliances such as air conditioner and

fans to obtain smooth speed control and achieve significant energy savings [120]. For an appliance with VFD, its power consumption for different scheduling intervals is different while the power consumption for a single scheduling interval τ is the same. For an appliance without VFD, its power consumption in different scheduling intervals remains the same. The above energy consumption scheduling formulation handles the household appliances with VFD. That is, an appliance does not change operating frequency within an interval but can operate at different frequencies in different scheduling intervals.

5.3.2 Energy Adaptation Variable based Offline Stochastic Scheduling

In this paper, motivated from [104], a systematic trip rate-driven stochastic scheduling algorithm is proposed to derive the desired energy adaptation variable β and generate the operation schedule for a given set of household appliances with trip rate requirements. Note that this is an offline scheduling (to handle the uncertainty in household appliances) and the online scheduling scheme is described in Section 5.3.3 (to handle the uncertainty in renewable generation).

As described in Fig. 5.3, for a given set of appliances $a \in A$ and the target trip rate ρ_t , a value of the energy adaptation variable β is iteratively picked, and the actual energy consumed by each

appliance in the scheduling horizon T is derived using Equation (5.2) based on the selected β (step A). The operation schedule of the appliance set is then generated through solving the linear program (step B). Finally, the trip rate ρ of the resultant operation schedule is derived using Monte Carlo simulation (step C). If the trip rate of the schedule does not satisfy the stop-condition of the algorithm, the energy adaptation variable β is adjusted and the above process is repeated. If the trip rate of the operation schedule satisfies the stop-condition of the algorithm, the valid operation schedule is generated and its trip rate meets the system trip rate requirement. In other words, the output operation schedule is the desired operation schedule if $(\rho_t - \rho) \geq \varepsilon > 0$ holds, where ε is an arbitrarily small positive number. The design flow of the algorithm is

illustrated in Figure 5.4. Each step of the offline scheduling algorithm is described in details in the following subsections.

5.3.2.1 β -Enabled Parallel Appliance Operation Scheduling

The energy adaptation variable β can adapt energy consumed by an appliance to the Gaussian probability distribution of energy consumption patterns. It enables the trip rate-driven scheduling of appliance operation based on pricing information released ahead of time. Since $0 \leq \beta \leq 1$, the energy consumed by an appliance $a \in A$ during the scheduling horizon T ranges

Input: γ_a^{min} and γ_a^{max} for $a \in A$; target trip rate ρ_t

Output: the desired β ; expense efficient operation schedule

1. pick a value of β in $0 \leq \beta \leq 1$
2. **repeat**
3. A : update E_a^T for $a \in A$ based on β
4. B : generate an operation schedule by solving the LP
5. C : derive ρ for the operation schedule using Monte Carlo simulation; update β
6. **until** $(\rho_t - \rho) \geq \varepsilon > 0$

Figure 5.3: The Offline stochastic scheduling algorithm to iteratively derive the energy adaptation variable β and generate the monetary expense efficient operation schedule

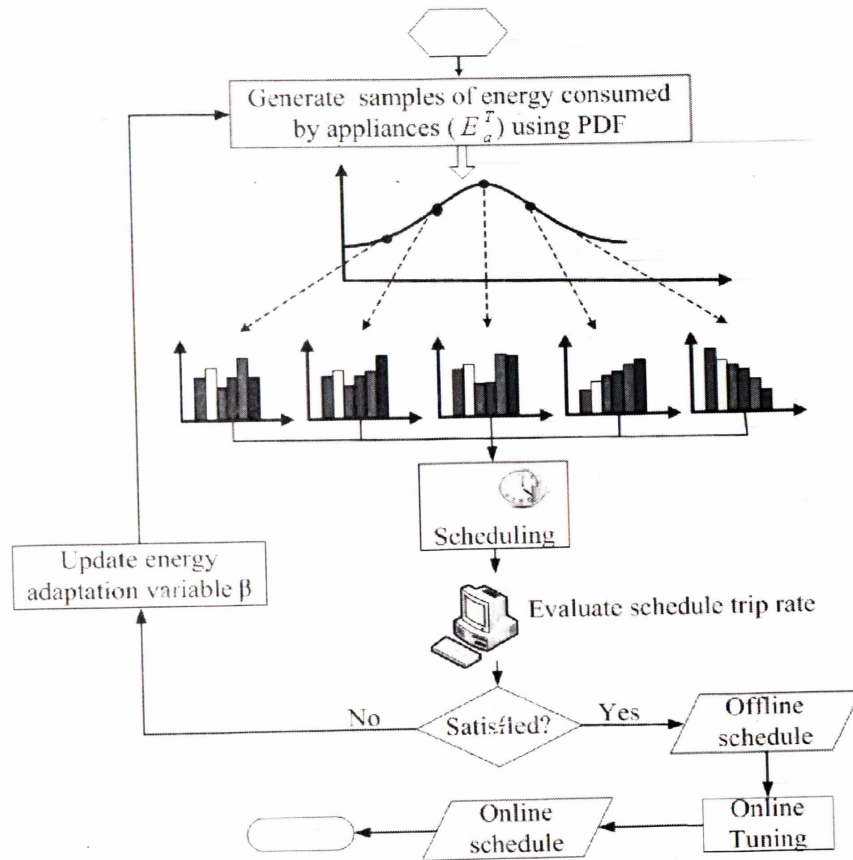


Figure 5.4: The design flow of the household appliance energy consumption scheduling algorithm

from γ_a^{\min} to γ_a^{\max} , as is shown in Equation (5.2).

Due to the statistical property of energy consumption patterns, there exists no fixed relationship between the energy adaptation variable β and the trip rate of a schedule. Although the

current trip rate ρ of the operation schedule can not be directly used to find the value of β in the next iteration, the energy consumed by a household appliance is linear with the energy adaptation variable β according to Equation (5.2). Therefore, it is natural to use a step search strategy to derive the desired β that minimize the customer monetary expense. The β is initialized to 0 and the step length depends upon the time requirements to generate the desired operation schedule.

The introduction of the energy adaptation variable β in fact makes the proposed algorithm parallelization friendly through utilizing the powerful computing capacity of modern computers. The iterative appliance operation scheduling algorithm shown in Fig. 5.3 is essentially a pipelined processing stages, each of which consists of using β to adapt

energy consumed by an appliance to energy consumption patterns, solving linear program to generate an operation schedule, and evaluating the resultant operation schedule. This pipelining property of the algorithm naturally facilitates the concurrent generation of operation schedules and strikingly reduces the time to generate the desired β and operation schedule. The key fact is that the above procedure with different β can be performed independently. For example, for a given a dual core processor, if the current value of β is β_0 and current trip rate does not meet the specified stop-condition, the next iterations of the algorithm with $\beta = \beta_0 + \xi$ and $\beta = \beta_0 + 2\xi$ can be run simultaneously on core 1 and core 2, respectively, where ξ is the specified step length.

5.3.2.2 Derive the Trip Rate Using Monte Carlo Simulation and Latin-Hypercube Sampling

The trip rate of an operation schedule for a given set of appliances is evaluated using Monte Carlo simulation under the assumption that the energy consumption patterns follows the Gaussian probability distribution. Oftentimes, the trip rate of an operation schedule is obtained in two major steps. In the first step, the energy consumed by an appliance during the scheduling horizon is generated based on the probability distribution of the energy consumption patterns. In the second step, the summation of the energy consumption by household unit is derived, and whether the operation schedule trips out is verified.

The first step is the Monte Carlo sample generation and the second step is the Monte Carlo sample evaluation. Repeat the two steps to take a sufficiently large number of (say 10,000) Monte Carlo samples, and the trip rate of the current operation schedule can be estimated as the ratio of the number of samples where the system trips out to the total number of Monte Carlo samples. If the current trip rate is less than and yet close enough to the target trip rate, the resultant operation schedule is the desired schedule and the execution of the algorithm exits. Otherwise, the algorithm jumps from step *C* to step *A* and continues its execution, as is described in Fig. 5.3.

Although the Monte Carlo simulation described above exhibits relative generality and insensitivity to stochastic characteristics of energy consumption patterns, it is expensive for accurate trip

rate estimation of an operation schedule. Therefore, the Latin Hypercube sampling method is adopted in this work to improve the efficiency of trip rate estimation for an operation schedule by sampling energy consumption more systematically.

The Latin Hypercube sampling (LH) [68] has been utilized in uncertainty analysis to generate multivariate samples of statistical distributions. For a one-dimensional variable-sampling, it starts by estimating the uncertainty of a variable using a probability distribution, using identical probability windows, and generating a sample value for the variable in each interval.

A two-dimensional variable-sampling is used to illustrate the idea of LH sampling. Given two random variables (x,y) that

are to be simulated. 4 simulation samples of the variable can be obtained by randomly generating 4 pairs of (x,y) . As is shown in Fig. 5.5(a), these samples cannot well represent the simulation space.

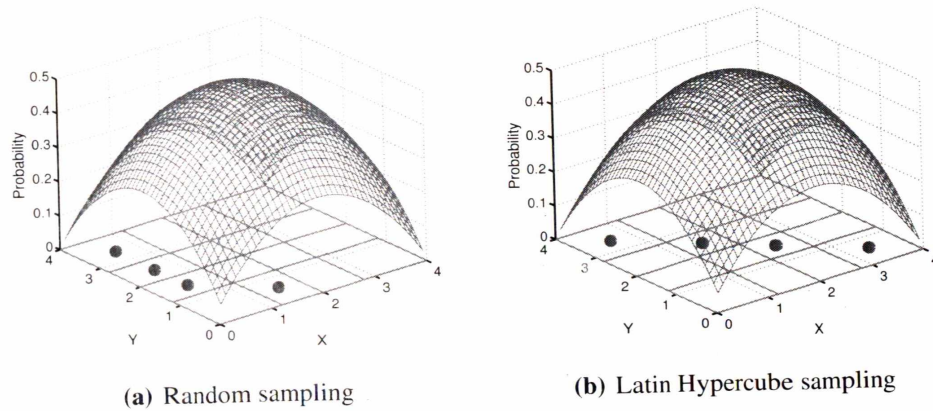


Figure 5.5: Random sampling versus Latin Hypercube sampling

LH sampling technique can be utilized to tackle the above non-representative sampling issue. In LH sampling, identical probability windows are used, each of which can associate with one and only one simulation sample. As is illustrated in Fig. 5.5(b), the simulation domain is first divided into 4×4 grids

of equal probability. When one grid is selected to generate a simulation sample, grids with the same row or column can not be the candidate grids for future sample generation. The samples in Fig. 5.5(a) represent the standard Monte Carlo samples. They cover only a small part of the simulation space, and thus a large number of samples are needed. In contrast, LH sampling can use much smaller number of samples to cover the simulation space. As a result, the simulation time can be significantly reduced.

5.3.3 Adaptation of Offline Operation Schedules to Online Scheduling

The proposed energy consumption scheduling algorithm consists of two parts, that is, the offline scheduling algorithm and the online adaptation algorithm. The offline scheduling

algorithm is first designed assuming that all inputs of the algorithm are given. In other words, it is assumed that the energy consumed by household appliances and the energy produced by the solar panel are known in advance. As a result, the offline operation schedule is optimum. However, when the system is in operation, the energy consumed by household appliances and the energy produced by the solar panel deviate from the values utilized to optimize the offline operation schedule. Thus, the optimality of the offline operation scheduling is lost and the online operation needs to be tuned to compensate for the optimality loss.

Let $\Delta \sum_{a \in A} x_a^\tau$ denote the runtime variation (increase) in the energy demand of appliances $a \in A$ in the interval of τ , and let Δe_s^τ the runtime variation (increase) in the harvested solar

energy, then the combined variation (increase) in grid energy attributed to the concerned household appliances and the solar energy is denoted by ΔP^τ and can be expressed as $\Delta P^\tau = \Delta e_s^\tau - \Delta \sum_{a \in A} x_a^\tau$. The ΔP^τ is essentially the modified variation in the solar energy by considering the offset to the variation in energy demand of household appliances. The ΔP^τ could be greater than, less than, or equal to 0. For each case, the battery status is checked before disposal of solar energy. The solar energy could be consumed, stored in the battery for future use, or wasted due to limitation of battery capacity, as is detailed as follows.

† $\Delta P^\tau > 0$: If battery is full, and the consumption of energy from electrical grid is not scheduled, that is, only solar energy is scheduled at the moment, then discard the extra

solar energy; if battery is full and the consumption of energy from electrical grid is scheduled, then use solar instead of utility energy. If battery is not full, and the current price of utility electricity is higher than the average utility electricity price, then use the solar energy; else if the current electricity price is lower than the average electricity price, then store the solar energy to battery.

† $\Delta P^r < 0$: If battery is empty, then use the energy from electrical grid. If battery is not empty and the current utility electricity price is higher than the average utility electricity price, then use batter energy, else use the utility energy.

† $\Delta P^r = 0$: Follow the offline appliance operation schedule.

It is worth noting that the online tuning algorithm is an integral

part of the proposed scheduling scheme. The uncertainty in the energy demand of household appliances has been handled by the offline scheduling algorithm, thus, the runtime variation in the energy demand has minimal impact to the scheduling results. With respect to the solar energy, comparing to the total energy consumption, it is only a very small portion of the. Therefore, the runtime variation of the solar energy is much smaller and its impact to the scheduling results is negligible. In fact, the online tuning algorithm aims at maximizing the benefit of the solar energy instead of dealing with the impact of the above variations to the scheduling results. The online algorithm prioritizes the solar energy and the energy from the grid, and utilizes solar energy at a higher priority.

Table 5.2
Comparison of the deterministic LP-based approach and the traditional scheduling approach

Appliance set size	Traditional		Deterministic LP-based		
	Runtime(s)	Expense(€)	Runtime(s)	Expense(€)	Expense reduction
5-10	0.02	102.7	0.8	60.0	41.7%
11-20	0.03	220.5	1.2	128.4	41.8%
16-20	0.05	294.5	1.8	174.6	40.7%
21-25	0.08	330.0	2.3	185.7	43.7%
26-30	0.11	393.6	2.9	220.5	44.1%

Table 5.3
Comparison of the worst case, the best case, and the stochastic design in terms of monetary expense and scheduling time

Taskset size	Worst case design ($\beta = 1$)			Best case design ($\beta = 0$)			Proposed stochastic design ($0 < \beta < 1$)			
	Trip rate	Run time(s)	Expense(€)	Trip rate	Run time(s)	Expense(€)	Trip rate	Run time(s)	Expense(€)	\$ reduction
5-10	0%	0.7	79.0	3.8%	0.7	41.7	0.5%	2.8	60.7	23.2%
11-20	0%	1.1	172.4	15.9%	1.2	86.5	0.5%	3.5	149.7	13.2%
16-20	0%	1.8	238.3	25.1%	1.8	116.1	0.5%	7.6	211.4	11.3%
21-25	0%	2.3	256.0	29.5%	2.4	121.4	0.5%	8.8	226.3	11.6%
26-30	0%	2.8	301.0	39.0%	2.8	143.8	0.5%	9.6	270.8	10.0%

Table 5.4
Comparison of the proposed stochastic designing approach with the traditional scheduling approach

Appliance set size	Traditional		Proposed stochastic design		
	Runtime(s)	Expense(€)	Runtime(s)	Expense(€)	Expense reduction
5-10	0.02	102.7	2.8	60.7	40.9%
11-20	0.03	220.5	3.5	149.7	32.1%
16-20	0.05	294.5	7.6	211.4	28.2%
21-25	0.08	330.0	8.8	226.3	31.4%
26-30	0.11	393.6	9.6	270.8	31.2%

5.4 Simulation Results and Discussions

Experiments simulation have been performed to demonstrate the our algorithm, which aims to reduce customer monetary expenses by utilizing the grid energy at off-peak times and maximizing the benefit from the solar energy. Sets of household appliances are carefully designed and generated. The number of household appliances is from 5 to 30, which is the typical number of household appliances [121]. The operation start time δ_a and end time θ_a of an appliance $a \in A$ are typically defined by customers. These values are such designed and generated in the scheduling horizon that they conform to practice pattern of human being. The scheduling horizon is assumed to be 24 hours. It is assumed that the deviation and

mean value of the energy consumption of an appliance during the scheduling horizon are known in advance. The mean of the energy consumed by an appliance is generated according to power characteristics of the appliance, and the standard deviation is set to 20% of the μ which is the mean. Hence, the maximum energy value γ_a^{max} and minimum energy value γ_a^{min} of the appliance $a \in A$ during the scheduling horizon T can be derived based on the given mean and standard deviation. The one day ahead pricing data released by Ameren Illinois Power Corporation [103] are available online, and are taken as the price input in the experiment.

In the simulation, two sets of the KD200-54 P series PV modules from the Kyocera Solar Incorporation [122] are taken to construct a solar station for a residential unit. The total cost

of the two sets is 502 \$ [123]. Assuming the lifetime of the PV system is 20 years [124], the PV installation cost normalized to 24 hours in the experiment, is $I_c = 0.055$ \$. A battery of 845 kW throughput is taken as energy storage. The battery costs 75 \$, thus, the cost per kW is $b_u = 0.089$ \$ [125].

The peak power of the module in the data sheet is 220W. For the offline scheduling, the energy produced by the solar station in each hour from 10:00 AM to 18:00 PM is assumed to be 100Wh, 210Wh, 300Wh, 400Wh, 400Wh, 300Wh, 210Wh, and 100Wh, respectively. The solar energy e_s^τ produced in the interval of τ could be derived from these data. The solar operation and maintenance cost is assumed to be $c_s^\tau = 0.01$ \$ per kWh, which is mostly due to washing the modules to remove dirt and dust instead of due to the rare occurrence of technical

failure [126]. In the runtime, the generated solar energy is assumed to deviate from the energy produced offline by 6%.

The proposed scheme was implemented in C++ and tested on a Pentium Dual Core machine with 2.3 GHz T4500 CPU and 3GB main memory. Simulation was run 1000 times on each set of appliances, and the reported results are the results averaged over all the runs.

Two approaches, referred to as the traditional scheduling approach and the deterministic LP-based approach, are compared in scheduling time and monetary expense of energy.

The traditional scheduling approach represents a typical scenario of energy consumption pattern for the traditional grid while the deterministic LP-based scheme is the basis for the

proposed scheduling algorithm.

In the traditional scheduling approach, an appliance is switched on by the customer whenever it is needed and switched off when the customer finishes using it. For simplicity, it is assumed that the appliance operates at its rated power. In this simulation of the experiment, the appliance $a \in A$ with the feasible interval of $[\delta_a, \theta_a]$ is supposed to be switched on at the instance δ_a , operate at its rated power, and is switched off before the instance θ_a . This arrangement of the appliance operation does not consider the time-varying electricity price in the execution duration of the appliance, thus incurring high customer monetary expenses.

In the deterministic LP-based approach, the operation schedule

is generated by running the LP solver only once other than running the LP solver iteratively. As is shown in TABLE 5.2, for a residential unit with different number of household appliances, the deterministic LP-based approach achieves significant savings while taking longer scheduling time. For instance, for a household with 5-10 appliances, the monetary expense of energy for the traditional scheduling approach and the deterministic LP-based approach is 102.7 and 60.0, respectively, while the scheduling time of the two schemes is 0.02s and 0.8s, respectively. The deterministic LP-based approach reduces the monetary expense of energy by about 42%.

Three cases of β -based energy consumption scheduling schemes, that is, the worst case ($\beta = 1$), the best case

$(\beta = 0)$, and the proposed stochastic case $(0 < \beta < 1)$, were implemented. In the worst case where an appliance $a \in A$ is assumed to consume the maximum amount of energy γ_a^{max} , the operation schedule is generated by running the LP solver only once. Similarly, in the best case, the LP solver takes as input the minimum energy value γ_a^{min} for each appliance $a \in A$, and runs only once to produce the operation schedule. On the contrary, the proposed stochastic design approach takes as input the β -based energy budget for each appliance $a \in A$, and runs the LP solver iteratively to generate an expense efficient operation schedule under a given trip rate requirement.

The three design cases are then compared in the expense of energy consumption and scheduling time. The scheduling time of the worst case and best case design is less than that of the

stochastic design. This is because the worst case and the best case designing approaches run the LP solver only once while the proposed stochastic approach iteratively runs the LP solver to generate the operation schedule. The monetary expense of the worst case design is greater than that of the stochastic design, which is in turn greater than that of the best case design. This is because that the worst case design overestimates the energy consumed by individual appliances while the best case design underestimates the energy consumption. This misestimate of the energy consumed by individual appliances also leads to the discrepancy in trip rate of different designing approaches. Although the best case solution has the minimum monetary expense, it leads to significant tripping out which is useless in practice. As is illustrated in TABLE 5.3, for a

residential unit with 16-20 household appliances and with trip rate requirement of 0.5%, the scheduling time of the worst case, the best case, and the proposed stochastic design is 1.8s, 1.8s, and 7.6s, respectively, and the expense of utility energy of the three designs is 238.3, 116.1, and 211.4, respectively. Due to the overestimation and underestimation of the energy consumed by individual appliance in the residential unit, the trip rate of the worst case and the best case design is 0% and 25.1%, respectively. The trip rate of the proposed stochastic design is 0.5% for the residential unit, which satisfies the given trip rate requirement.

The traditional scheduling and the stochastic scheduling approach are compared in terms of monetary expenses and scheduling time, and the results are presented in TABLE 5.4.

The proposed stochastic design achieves up to 41% reduction in monetary expenses. The proposed scheduling algorithm can always generate a monetary expense efficient operation schedule within 10 seconds, which is fast enough for household appliance applications.

Due to uncertainty of the renewable energy source and the variation of energy demand of household appliances, the offline operation schedule need to be tuned to fully utilize the benefit from the solar energy and guarantee the customer satisfaction without violating the system trip rate requirement. Figure 5.6 shows a comparison of the customer monetary expense for online and offline operation schedule. For various number of household appliances, the customer monetary expense of online operation schedule is close to that of the optimized

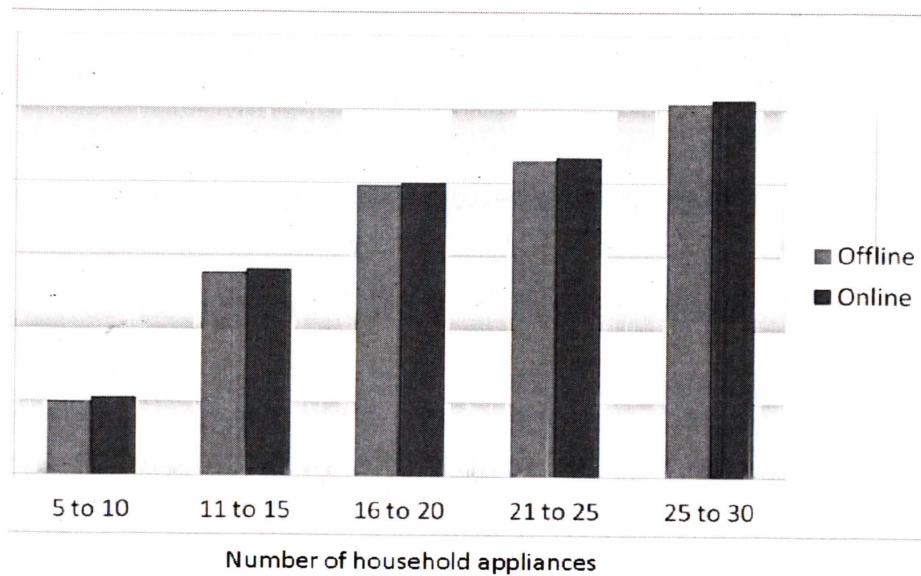


Figure 5.6: Compare the monetary expense of online and offline operation schedule
offline operation schedule.

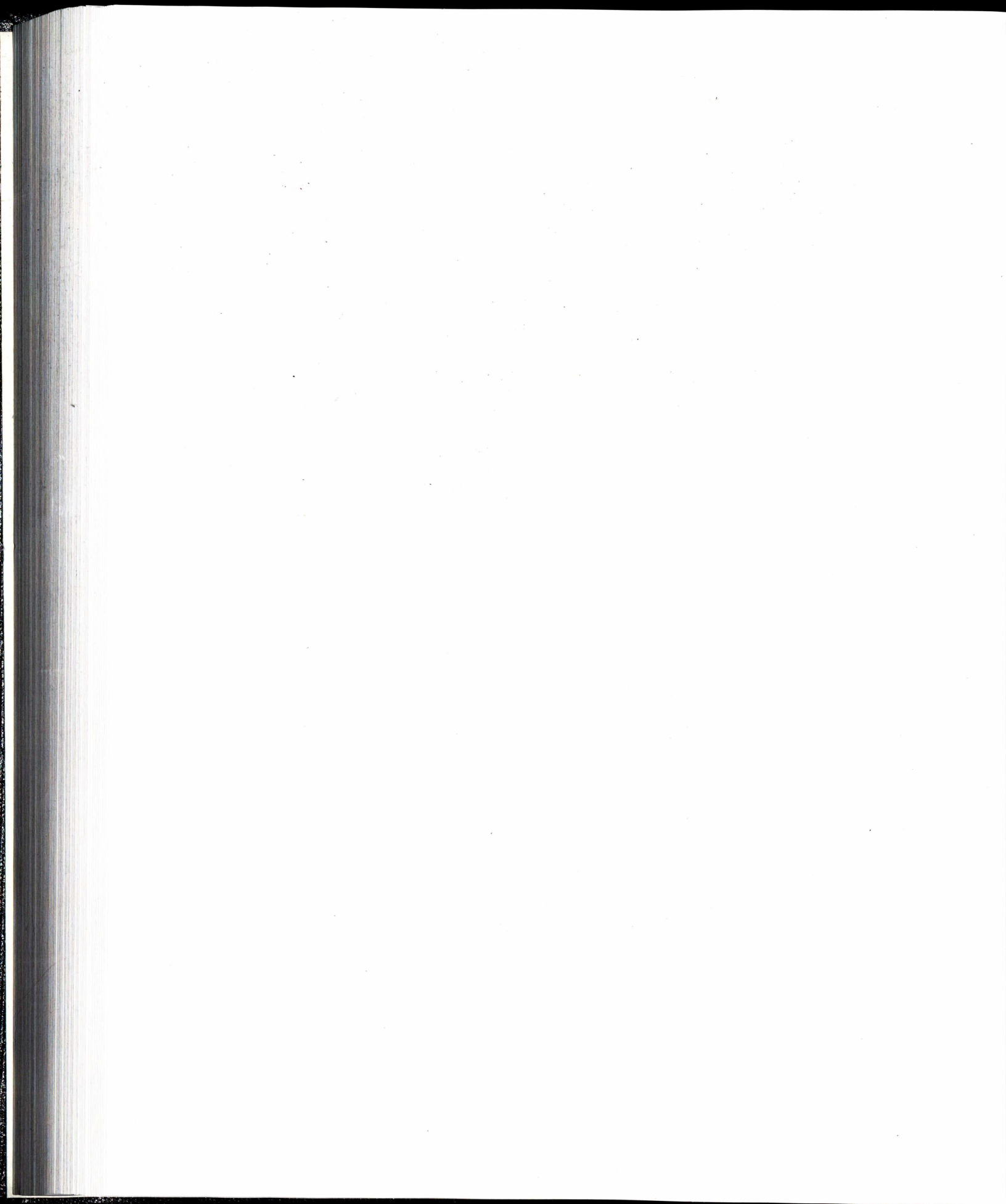
5.5 Conclusions

This paper proposes a stochastic energy consumption scheduling algorithm based on the time-varying pricing information released by utility companies ahead of time. The

proposed energy consumption scheduling algorithm is featured by an energy adaptation variable β that models the stochastic property of customer energy consumption practices. It takes as input the minimum and maximum amount of energy consumed by individual appliances and the pre-defined target trip rate. The output of the proposed algorithm is the desired β that approximates the probability distribution of customer energy consumption practice, and the expense efficient appliance energy consumption schedule. The proposed appliance operation scheduling algorithm also accelerates the generation of the desired operation schedule by paralleling the computing process. Experimental simulation demonstrates that the proposed algorithm reduce monetary expenses up to 41% when compared to the traditional scheduling scheme that models

typical appliance operations in traditional home scenario. The results also demonstrate that when compared to a worst case design where an appliance is assumed to consume the maximum amount of energy, the proposed design that considers the stochastic energy consumption patterns achieves up to 24% monetary expenses reduction without violating the target trip rate of 0.5%. The monetary expense of the runtime operation schedule is close to that of the offline optimized operation schedule. Furthermore, the proposed scheduling algorithm can always generate a monetary expense efficient operation schedule within 10 seconds, which is fast enough for household appliance applications. The future work seeks to investigate the game theory based scheduling to reduce peak-to-average ratio for a smart community based on the scheduling technique

proposed in this paper.



Chapter 6

Conclusions

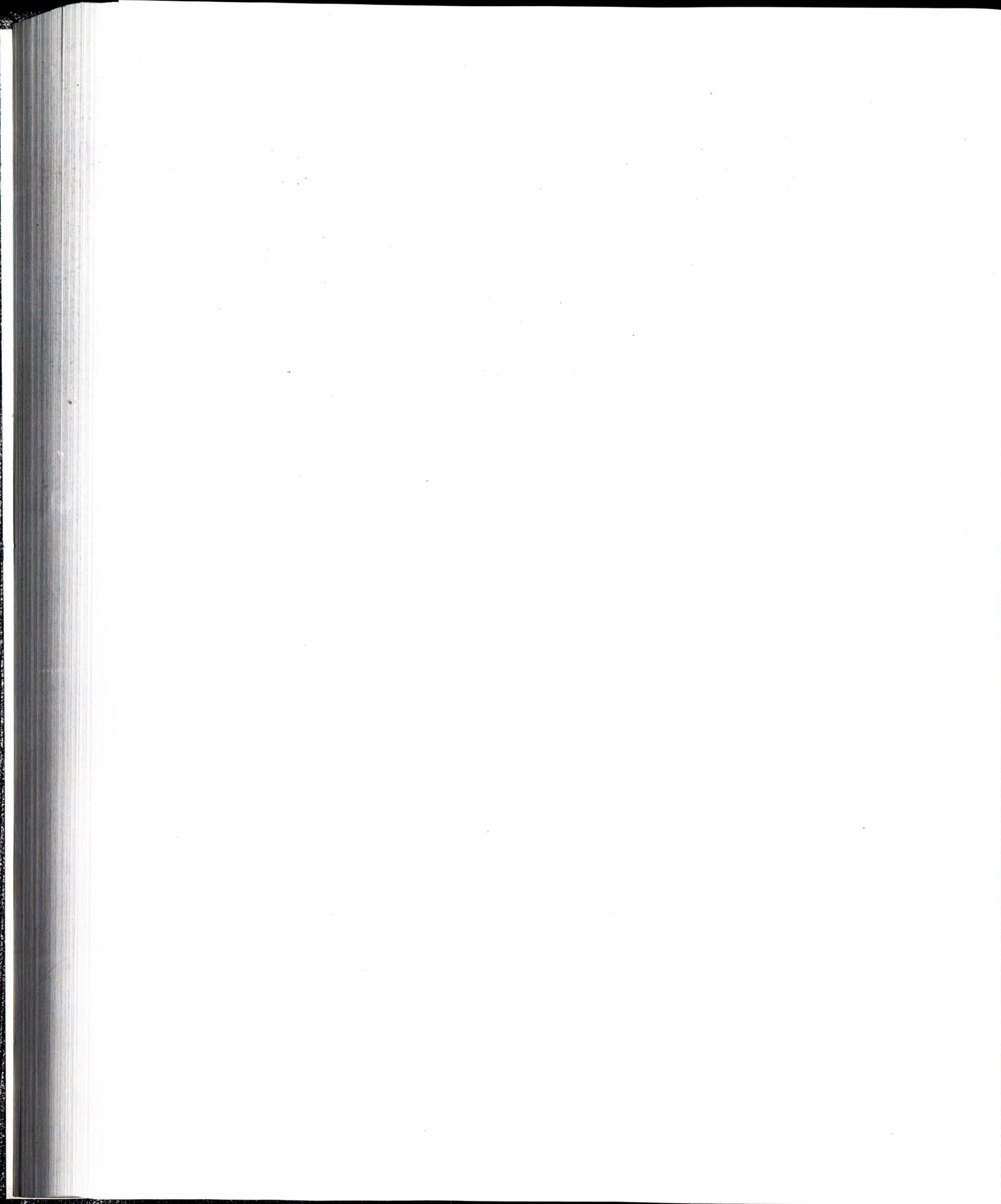
In this dissertation, several innovative algorithmic CAD techniques are proposed for VLSI circuit design, and smart home embedded system Design.

In the part of VLSI circuit design, reliability interconnect and the optical interconnect are addressed in this dissertation. For reliability interconnect, a new reliability-driven minimum cost Steiner routing and layer assignment approach is proposed.

Our new algorithm is based on the integer linear programming formulation. This allows to compute the optimal routing on small nets, and it greatly improves the solution quality on large nets with up to 14.7% wire reduction compared to the state-of-the-art schemes presented in [1]. In addition, our new algorithm can save 11.4% wires over a heuristic algorithm for handling multiple currents. For the optical interconnect, the dissertation proposes the first transceiver insertion algorithmic framework which can reduce the energy demand by $2.6\times$ compared to a natural minimum spanning tree heuristic and improve the timing by about $2\times$ compared to copper trees.

In the part of embedded system, a reliability-driven task scheduling scheme is proposed for multiprocessor real-time embedded systems that optimizes system energy consumption

under stochastic fault occurrences. With this scheme, the energy saving is more than 15% compared to the approach of designing for the corner case of fault occurrences. For the smart home embedded system, a new energy efficient scheduling algorithm is proposed to arrange the household appliances. Compared to a natural greedy algorithm which models typical household appliance operations in the traditional home scenario, the proposed deterministic linear programming based scheduling scheme achieves up to 53% monetary expense reduction. Compared to a worst case design where an appliance is assumed to consume the maximum amount of energy, the proposed stochastic design which considers the stochastic energy consumption patterns achieves up to 24% monetary expense reduction without violating the target trip rate of 0.5%.



References

- [1] J. Lienig and G. Jerke, "Current-driven wire planning for electromigration avoidance in analog circuits," in *Proceedings of the 2003 Asia and South Pacific Design Automation Conference*, ser. ASP-DAC '03, 2003, pp. 783–788.
- [2] G. Chen, H. Chen, M. Haurylau, N. Nelson, D. Albonesi, P. Fauchet, and E. Friedman, "On-chip copper-based vs. optical interconnects: Delay uncertainty, latency, power, and bandwidth density comparative

predictions,” *Interconnect Technology Conference, 2006 International*, pp. 39–41, 2006.

- [3] S. S. and F. A. (2011, Jun.) Online resource of dynamic pricing. [Online]. Available: http://brattle.com/_documents/UploadLibrary/Upload956.pdf
- [4] IRTS, *International Technology Roadmap For Semiconductors*. Semiconductor Industry Association, 2005, vol. California.
- [5] J. Cong and Y. Zhang, “Thermal-aware physical design flow for 3-d ics,” in *Proceedings of the 23rd International VLSI Multilevel Interconnection Conference (VMIC)*, 2006, pp. 73–80.
- [6] S. K. Reinhardt and S. S. Mukherjee, “Transient fault

detection via simultaneous multithreading,” *SIGARCH Comput. Archit. News*, vol. 28, no. 2, pp. 25–36, 2000.

- [7] J. Black, “Electromigration - a brief survey and some recent results,” in *Proceedings of IEEE International Reliability Physics Symposium*, vol. ED-16, no. 4, 1968, pp. 338–347.
- [8] S. Hu, Z. Li, and C. J. Alpert, “A polynomial time approximation scheme for timing constrained minimum cost layer assignment,” in *Proceedings of the 2008 IEEE/ACM International Conference on Computer-Aided Design*, ser. ICCAD '08, 2008, pp. 112–115.
- [9] J. Lienig, G. Jerke, and T. Adler, “Electromigration avoidance in analog circuits: Two methodologies for

current-driven routing,” in *Proceedings of the 2002 Asia and South Pacific Design Automation Conference*, ser. ASP-DAC '02, 2002, pp. 372–.

[10] G. Jerke, J. Lienig, and J. Scheible, “Reliability-driven layout decompaction for electromigration failure avoidance in complex mixed-signal ic designs,” in *Proceedings of the 41st annual Design Automation Conference*, ser. DAC '04, 2004, pp. 181–184.

[11] G. Jerke and J. Lienig, “Hierarchical current-density verification in arbitrarily shaped metallization patterns of analog circuits,” *IEEE Trans. on Comp.-Aided Des. Integ. Cir. Sys.*, vol. 23, no. 1, pp. 80–90, Nov. 2006.

[12] —, “Hierarchical current density verification for electromigration analysis in arbitrary shaped

metallization patterns of analog circuits,” in *Proceedings of the conference on Design, automation and test in Europe*, ser. DATE '02, 2002, pp. 464–450.

- [13] J. Lienig, “introduction to electromigration-aware physical design,” in *Proceedings of the 2006 international symposium on Physical design*, ser. ISPD '06, 2006, pp. 39–46.
- [14] M. R. Garey and D. S. Johnson, *Computers and Intractability; A Guide to the Theory of NP-Completeness*. New York, NY, USA: W. H. Freeman & Co., 1990.
- [15] L. Behjat and A. Chiang, “Fast integer linear programming based models for vlsi global routing.” in *ISCAS (6)*. IEEE, 2005, pp. 6238–6243.

- [16] T. Terlaky, A. Vannelli, and H. Zhang, "On routing in VLSI design and communication networks," *Discrete Applied Mathematics*, vol. 156, no. 11, pp. 2178–2194, 2008.
- [17] S. Shah, A. Srivastava, D. Sharma, D. Sylvester, D. Blaauw, and V. Zolotov, "Discrete vt assignment and gate sizing using a self-snapping continuous formulation," in *Proceedings of the 2005 IEEE/ACM International conference on Computer-aided design*, ser. ICCAD '05, 2005, pp. 705–712.
- [18] M. Haurylau, G. Chen, H. Chen, J. Zhang, N. Nelson, D. Albonesi, E. Friedman, and P. Fauchet, "On-chip optical interconnect roadmap: Challenges and critical directions," *IEEE Journal of Selected Topics in Quantum*

Electronics, vol. 12, no. 6, pp. 1699–1705, 2006.

- [19] A. Krishnamoorthy and D. Miller, “Scaling Optoelectronic-VLSI Circuits into the 21st Century: A Technology Roadmap,” *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 2, pp. 55–76, 1996.
- [20] S. Palermo, A. Emami-Neyestanak, and M. Horowitz, “A 90 nm cmos 16 gb/s transceiver for optical interconnects,” *IEEE journal of solid-state circuits*, vol. 43, no. 5, pp. 1235–1246, 2008.
- [21] N. Sherwood-Droz, H. Wang, L. Chen, B. Lee, A. Biberman, K. Bergman, and M. Lipson, “Optical 4x4 hitless silicon router for optical networks-on-chip (noc),” *Optics Express*, vol. 16, no. 2, pp. 15 915–15 922, 2008.

- [22] S. Bahirat and S. Pasricha, "Exploring hybrid photonic networks-on-chip foremerging chip multiprocessors," in *Proceedings of the 7th IEEE/ACM international conference on Hardware/software codesign and system synthesis*, ser. CODES+ISSS '09, 2009, pp. 129–136.
- [23] H. Yamada, M. Nozawa, M. Kinoshita, and K. Ohashi, "Vertical-coupling optical interface for on-chip optical interconnection," *Optics Express*, vol. 19, no. 6, pp. 698–703, 2011.
- [24] C. R., "Optical interconnects and vlsi photonics," *IEEE LEOS Summer Topical Meeting, LEOS Newsletters*, no. 5, pp. 8–9, 2004.
- [25] K. M., "On-chip optical interconnects," *Intel Technol.*, vol. J8, no. 2, pp. 129–141, 2004.

- [26] D. Miller, "Device requirement for optical interconnects to silicon chips," *Proceedings of IEEE, Special Issue on Silicon Photonics*, vol. 97, no. 7, pp. 1166–1185, 2009.
- [27] I. Conor and F. Gaffiot, "On-chip optical interconnect for low-power," *Ultra-Low Power Electronics and Design*, pp. 17–26, 2004.
- [28] Y. Vlasov, "Silicon photonics for next generation computing systems," *Optical Communication, 2008. ECOC 2008. 34th European Conference on*, pp. 1–2, 2008.
- [29] G. Chen, H. Chen, M. Haurylau, N. A. Nelson, D. H. Albonesi, P. M. Fauchet, and E. G. Friedman, "Predictions of cmos compatible on-chip optical

interconnect,” *Integr. VLSI J.*, vol. 40, no. 4, pp. 434–446, Jun. 2007.

- [30] D. Ding, Y. Zhang, H. Huang, R. T. Chen, and D. Z. Pan, “O-router: an optical routing framework for low power on-chip silicon nano-photonic integration,” in *Proceedings of the 46th Annual Design Automation Conference*, ser. DAC '09, 2009, pp. 264–269.
- [31] Y. Massoud, “Subwavelength nanophotonics for future interconnects and architectures,” *Invited Talk at NRI SWAN Center, Rice University*, 2008.
- [32] S. McNab, N. Moll, and Y. Vlasov, “Ultra-low loss photonic integrated circuit with membrane-type photonic crystal waveguides,” *Optical Express*, vol. 11, no. 22, pp. 2927–2939, 2003.

- [33] H. Rong, R. Jones, A. Liu, O. Cohen, D. Hak, A. Fang, and M. Paniccia, "A continuous-wave raman silicon laser," *Nature*, vol. 433, pp. 725–728, 2005.
- [34] S. Koester, G. Dehlinger, J. Schaub, J. Chu, Q. Ouyang, and A. Gril, "Germanium-on-insulator photodetectors," *IEEE International Conference on Group VI Photonics*, pp. 171–173, 2005.
- [35] M. Hochberg, T. Baehr-Jones, C. Walker, and A. Scherer, "Integrated plasmon and dielectric waveguides," *Optics Express*, vol. 12, pp. 5481–5486, 2004.
- [36] P. Pauzauskie, D. Sirbuly, and P.-D. Yang, "Semiconductor Nanowire Ring Resonator," *Physical Review Letters PRL* 96, 143903, 2006.

- [37] X. Dou, X. Wang, X. Lin, D. Ding, D. Pan, and R. Chen, "Highly flexible polymeric optical waveguide for out-of-plane optical interconnects," *Optics Express*, vol. 18, no. 15, pp. 16 227–16 233, 2010.
- [38] ———, "Polymeric waveguides with embedded micro-mirrors formed by metallic hard mold," *Optics Express*, vol. 18, no. 8, pp. 378–385, 2010.
- [39] D. Ding and D. Z. Pan, "Oil: a nano-photonics optical interconnect library for a new photonic networks-on-chip architecture," in *Proceedings of the 11th international workshop on System level interconnect prediction*, ser. SLIP '09, 2009, pp. 11–18.
- [40] J. R. Minz, S. Thyagaraja, and S. K. Lim, "Optical routing for 3d system-on-package," in *Proceedings of the*

conference on Design, automation and test in Europe:

Proceedings, ser. DATE '06, 2006, pp. 337–338.

- [41] X. Chen and S. Hu, “A transceiver-aware routing framework for on-chip nanophotonic integration,” in *Proceedings of IEEE International Workshop on Emerging Technologies in VLSI CAD and Embedded Computing*, ser. VCEC '10, 2010, pp. 595–600.
- [42] L. van Ginneken, “Buffer placement in distributed rc-tree networks for minimal elmore delay,” in *Proceedings of IEEE International Symposium on Circuits and Systems*, 2010, pp. 865 – 868.
- [43] S. Hu, C. J. Alpert, J. Hu, S. K. Karandikar, Z. Li, W. Shi, and C. N. Sze, “Fast algorithms for slew-constrained minimum cost buffering,” *IEEE Trans. on Comp.-Aided*

Des. Integ. Cir. Sys., vol. 26, no. 11, pp. 2009–2022, Nov. 2007.

- [44] Y. Oh and S. Son, “Multiprocessor support for real-time fault-tolerant scheduling,” *IEEE Workshop on Architectural Aspect of Real-Time Systems*, 1991.
- [45] D. Mosse, R. Melhem, and S. Ghosh, “Analysis of a fault-tolerant multiprocessor scheduling algorithm,” in *IEEE Fault Tolerant Computing*, 1994, pp. 16–25.
- [46] S. Ghosh, R. Melhem, and D. Mossé, “Fault-tolerance through scheduling of aperiodic tasks in hard real-time multiprocessor systems,” *IEEE Trans. on Parallel Distrib. Syst.*, vol. 8, no. 3, pp. 272–284, Mar. 1997.
- [47] J.-J. Chen, H.-R. Hsu, K.-H. Chuang, C.-L. Yang, A.-C.

- Pang, and T.-W. Kuo, "Multiprocessor energy-efficient scheduling with task migration considerations," in *Proceedings of the 16th Euromicro Conference on Real-Time Systems*, ser. ECRTS '04, 2004, pp. 101–108.
- [48] C.-Y. Yang, J.-J. Chen, and T.-W. Kuo, "An approximation algorithm for energy-efficient scheduling on a chip multiprocessor," in *Proceedings of the conference on Design, Automation and Test in Europe - Volume 1*, ser. DATE '05, 2005, pp. 468–473.
- [49] J. Cong and K. Gururaj, "Energy efficient multiprocessor task scheduling under input-dependent variation," in *Proceedings of the Conference on Design, Automation and Test in Europe*, ser. DATE '09, 2009, pp. 411–416.
- [50] C. Xian, Y.-H. Lu, and Z. Li, "Energy-aware scheduling

for real-time multiprocessor systems with uncertain task execution time,” in *Proceedings of the 44th annual Design Automation Conference*, ser. DAC '07, 2007, pp. 664–669.

[51] R. Watanabe, M. Kondo, M. Imai, H. Nakamura, and T. Nanya, “Interactive presentation: Task scheduling under performance constraints for reducing the energy consumption of the gals multi-processor soc,” in *Proceedings of the conference on Design, automation and test in Europe*, ser. DATE '07, 2007, pp. 797–802.

[52] G. Zeng, T. Yokoyama, H. Tomiyama, and H. Takada, “Practical energy-aware scheduling for real-time multiprocessor systems,” in *Proceedings of the 2009 15th IEEE International Conference on Embedded and*

Real-Time Computing Systems and Applications, ser.
RTCSA '09, 2009, pp. 383–392.

- [53] D. Zhu, R. Melhem, and D. Mosse, “The effects of energy management on reliability in real-time embedded systems,” in *Proceedings of the 2004 IEEE/ACM International conference on Computer-aided design*, ser. ICCAD '04, 2004, pp. 35–40.

- [54] D. Pradhan, *Fault tolerance computing: Theory and techniques*. Prentice Hall, 1986.

- [55] R. Melhem, D. Mossé, and E. M. Elnozahy, “The interplay of power management and fault recovery in real-time systems,” *IEEE Trans. on Comput.*, vol. 53, no. 2, pp. 217–231, Feb. 2004.

- [56] Y. Zhang, K. Chakrabarty, and V. Swaminathan, "Energy-aware fault tolerance in fixed-priority real-time embedded systems," in *Proceedings of the 2003 IEEE/ACM international conference on Computer-aided design*, ser. ICCAD '03, 2003, pp. 209–.
- [57] Y. Zhang and K. Chakrabarty, "A unified approach for fault tolerance and dynamic power management in fixed-priority real-time embedded systems," *IEEE Trans. on Comp.-Aided Des. Integ. Cir. Sys.*, vol. 25, no. 1, pp. 111–125, Nov. 2006.
- [58] T. Wei, P. Mishra, K. Wu, and H. Liang, "Online task-scheduling for fault-tolerant low-energy real-time systems," in *Proceedings of the 2006 IEEE/ACM international conference on Computer-aided design*, ser.

ICCAD '06, 2006, pp. 522–527.

- [59] D. Zhu, H. Aydin, and J.-J. Chen, “Optimistic reliability aware energy management for real-time tasks with probabilistic execution times,” in *Proceedings of the 2008 Real-Time Systems Symposium*, ser. RTSS '08, 2008, pp. 313–322.
- [60] H. Aydin and D. Zhu, “Reliability-aware dynamic voltage scaling for energy-constrained real-time embedded systems,” in *Proceedings of the 26th IEEE International Conference on Computer Design*, ser. ICCD '08, Oct 2008, pp. 633–639.
- [61] T. Wei, P. Mishra, K. Wu, and H. Liang, “Fixed-priority allocation and scheduling for energy-efficient fault tolerance in hard real-time multiprocessor systems,”

IEEE Trans. on Parallel Distrib. Syst., vol. 19, no. 11,
pp. 1511–1526, Nov. 2008.

[62] M. W. Rashid, E. J. Tan, M. C. Huang, and D. H. Albonesi, “Power-efficient error tolerance in chip multiprocessors,” *IEEE Micro*, vol. 25, no. 6, pp. 60–70, Nov. 2005.

[63] P. Pop, K. H. Poulsen, V. Izosimov, and P. Eles, “Scheduling and voltage scaling for energy/reliability trade-offs in fault-tolerant time-triggered embedded systems,” in *Proceedings of the 5th IEEE/ACM international conference on Hardware/software codesign and system synthesis*, ser. CODES+ISSS '07, 2007, pp. 233–238.

[64] X. Qi, D. Zhu, and H. Aydin, “Global reliability-aware

power management for multiprocessor real-time systems,” in *Proceedings of the 2010 IEEE 16th International Conference on Embedded and Real-Time Computing Systems and Applications*, ser. RTCSA '10, 2010, pp. 183–192.

[65] R. Jejurikar, C. Pereira, and R. Gupta, “Leakage aware dynamic voltage scaling for real-time embedded systems,” in *Proceedings of the 41st annual Design Automation Conference*, ser. DAC '04, 2004, pp. 275–280.

[66] N. Weste and K. Eshraghian, *Principles of CMOS VLSI Design: A System Perspective*. Addison-Wesley Publishing Company, 1992.

[67] S. Shah, A. Srivastava, D. Sharma, D. Sylvester,

D. Blaauw, and V. Zolotov, "Discrete vt assignment and gate sizing using a self-snapping continuous formulation," in *Proceedings of the 2005 IEEE/ACM International conference on Computer-aided design*, ser. ICCAD '05, 2005, pp. 705–712.

[68] M. McKay, R. Beckman, and W. Conover, "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics (American Statistical Association)*, vol. 21, no. 2, pp. 239–245, 1979.

[69] L. Behjat, A. Vannelli, and W. Rosehart, "Integer linear programming models for global routing," *INFORMS J. on Computing*, vol. 18, no. 2, pp. 137–150, Jan. 2006.

[70] G. Venayagamoorthy, "Potentials and promises of

computational intelligence for smart grids,” *IEEE Power and Energy Society General Meeting*, pp. 1–6, 2009.

- [71] E. Lightner and S. Widergren, “An orderly transition to a transformed electricity system,” *IEEE Trans. on Smart Grid*, vol. 1, no. 1, pp. 3–10, 2010.
- [72] G. Masters, *Renewable and Efficient Electric Power Systems*. Wiley Online Library, 2004.
- [73] A. Mohsenian-Rad, V. Wong, J. Jatskevich, R. Schober, and A. Leon-Garcia, “Autonomous demand-side management based on game-theoretic energy consumption scheduling for the future smart grid,” *IEEE Trans. on Smart Grid*, vol. 1, no. 3, pp. 320–331, 2010.

- [74] A. Mohsenian-Rad, V. Wong, J. Jatskevich, and R. Schober, "Optimal and autonomous incentive-based energy consumption scheduling algorithm for smart grid," *Innovative Smart Grid Technologies (ISGT)*, vol. 1, no. 1, pp. 1–6, 2010.
- [75] A. Ipakchi and F. Albuyeh, "Grid of the future," *IEEE Power and Energy Magazine*, vol. 2, no. 1, pp. 52–62, 2009.
- [76] P. Palensky and D. Dietrich, "Demand side management: Demand response, intelligent energy systems, and smart loads," *IEEE Trans. on Industrial Informatics*, vol. 7, no. 3, pp. 381–388, 2011.
- [77] A. Khodaei, M. Shahidehpour, and S. Bahramiradh, "Scuc with hourly demand response considering

- intertemporal load characteristics,” *IEEE Trans. on Smart Grid*, vol. 2, no. 3, pp. 564–571, 2011.
- [78] S. H., B.-M. S. H., F. T., J. F. P., G. V. R., and W. E. T., “The effects of electricity pricing on PHEV competitiveness,” *Energy Police*, vol. 39, no. 2, pp. 1552–1556, 2011.
- [79] S. S., M. Pipattanasomporn, and S. Rahman, “Demand response as a load shaping tool in an intelligent grid with electric vehicles,” *IEEE Trans. on Power App. Syst.*, vol. 2, pp. 624–631, Dec. 2011.
- [80] —, “Grid integration of electric vehicles and demand response with customer choice,” *IEEE Trans. on Smart Grid*, vol. 3, pp. 543–550, 2012.

- [81] I. S. and S. F. C., "Physically based modeling of cold load pickup," *IEEE Trans. on Power App. Syst.*, vol. PAS-100, no. 9, pp. 4142–4150, Sep. 1981.
- [82] T. S. and L. W., "A residential consumer-centered load control strategy in real-time electricity pricing environment," in . *39th North American Power Symposium*, ser. NAPS '07, 2007, pp. 505–510.
- [83] A. Vojdani, "Smart integration," *IEEE Power and Energy Magazine*, vol. 6, no. 6, pp. 71–79, 2008.
- [84] L. Tsoukalas and R. Gao, "From smart grids to an energy internet: Assumptions, architectures, and requirements," *International Conference on Electric Utility Deregulation and Restructuring and Power Technologies*, pp. 94–98, 2008.

- [85] S. Darby, *The Effectiveness of Feedback on Energy Consumption*. Environmental Change Institute, University of Oxford, 2006.
- [86] M. Pipattanasomporn, H. Feroze, and S. Rahman, "Multi-agent systems in a distributed smart grid: Design and implementation," *Proceedings on IEEE PES Power Systems Conference and Exposition*, pp. 1–8, 2009.
- [87] K. Moslehi and R. Kumar, "A reliability perspective of the smart grid," *IEEE Trans. on Smart Grid*, vol. 1, no. 1, pp. 57–64, 2010.
- [88] B. Russell and C. Benner, "Intelligent systems for improved reliability and failure diagnosis in distribution systems," *IEEE Trans. on Smart Grid*, vol. 1, no. 1, pp. 48–56, 2010.

- [89] T. Kim and H. Poor, "Scheduling power consumption with price uncertainty," *IEEE Trans. on Smart Grid*, vol. 2, no. 3, pp. 519–527, 2011.
- [90] C. E. and J. Fuller, "Time-of-use pricing for load management programs in taiwan power company," *IEEE Trans. on Power System*, vol. 27, pp. 1170–1181, Aug. 2012.
- [91] A. Mohsenian-Rad and A. Leon-Garcia, "Optimal residential load control with price prediction in real-time electricity pricing environments," *IEEE Trans. on Smart Grid*, vol. 1, no. 2, pp. 120–134, 2010.
- [92] J. Lee, G. Park, M. Kang, H. Kwak, and S. Lee, "Design of a power scheduler based on the heuristic for preemptive appliances," *Intelligent Information And*

Database Systems, Lecture Notes In Computer Science,
vol. 6591/2011, no. 1, pp. 396–405, 2011.

- [93] J. Lee, G.-L. Park, S.-W. Kim, H.-J. Kim, and C. O. Sung, “Power consumption scheduling for peak load reduction in smart grid homes,” in *Proceedings of the 2011 ACM Symposium on Applied Computing*, ser. SAC ’11, 2011, pp. 584–588.
- [94] G. Xiong, C. Chen, S. Kishore, and A. Yener, “Smart (in-home) power scheduling for demand response on the smart grid,” *IEEE PES Innovative Smart Grid Technologies (ISGT)*, pp. 1–7, 2011.
- [95] S. Caron and G. Kesidis, “Incentive-based energy consumption scheduling algorithms for the smart grid,” in *Proceedings of IEEE International Conference on*

Smart Grid Communications (SmartGridComm), 2010,
pp. 391–396.

- [96] A. Molderink, V. Bakker, M. Bosman, J. Hurink, and G. Smit, “Management and control of domestic smart grid technology,” *IEEE Trans. on Smart Grid*, vol. 1, no. 2, pp. 109–119, 2010.
- [97] L. Wu, M. Shahidehpour, and T. Li, “Comparison of scenario-based and interval optimization approaches to stochastic scuc,” *IEEE Trans. on Smart Grid*, vol. 27, no. 2, pp. 913–921, 2012.
- [98] R. Kaewpuang, D. Niyato, and P. Wang, “Decomposition of stochastic power management for wireless base station in smart grid,” *IEEE Wireless Communications Letters*, vol. 1, no. 2, pp. 97–100, 2012.

- [99] L. Wu, M. Shahidehpour, and T. Li, "Stochastic security-constrained unit commitment," *IEEE Trans. on Power Systems*, vol. 22, no. 2, pp. 800–811, 2007.
- [100] —, "Cost of reliability analysis based on stochastic unit commitment," *IEEE Trans. on Power Systems*, vol. 23, no. 3, pp. 1364–1374, 2008.
- [101] L. Wu, M. Shahidehpour, and Y. Fu, "Security-constrained generation and transmission outage scheduling with uncertainties," *IEEE Trans. on Power Systems*, vol. 25, no. 3, pp. 1674–1685, 2010.
- [102] Wikipedia. (2011, Aug.) Online resource of renewable portfolio standard. [Online]. Available: http://en.wikipedia.org/wiki/Renewable_portfolio_standard

- [103] Ameren Services. (2011, Jun.) Online resource of real time electrical price. [Online]. Available: <https://www2.ameren.com>
- [104] T. Wei, X. Chen, and S. Hu, "Reliability-driven energy-efficient task scheduling for multiprocessor real-time systems," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 10, pp. 1569–1573, Oct. 2011.
- [105] H. Houthakker, "Electricity tariffs in theory and practice," *The Economic Journal*, vol. 61, no. 241, pp. 1–25, 1951.
- [106] P. Steiner, "Peak loads and efficient pricing," *Quarterly Journal of Economics*, vol. 71, no. 4, pp. 585–610, 1957.

- [107] S. Borenstein, "The long-run effects of real-time electricity pricing," *Center for the Study of Energy Markets - Working Paper*, 2004.
- [108] F. Wolak, "Residential customer response to real-time pricing: The anaheim critical peak pricing experiment," *Center for the Study of Energy Markets - Working Paper*, 2006.
- [109] W. Burke and D. Auslander, "Residential electricity auction with uniform pricing and cost constraints," *Proceedings of North American Power Symposium*, pp. 1-6, 2009.
- [110] P. Centolella, "The integration of price responsive demand into regional transmission organization (rto)

wholesale power markets and system operations,”
Energy, vol. 35, no. 4, pp. 1568–1574, 2010.

[111] B. Alexander, “Smart meters, real time pricing, and demand response programs: Implications for low income electric customers,” *Oak Ridge National Laboratory Technical Report*, 2007.

[112] S. Holland and E. Mansur, “Is real-time pricing green? the environmental impacts of electricity demand variance,” *Review of Economics and Statistics*, vol. 90, no. 3, pp. 550–561, 2008.

[113] J. Catalao, S. Marianoa, V. Mendesb, and L. Ferreira, “Short-term electricity prices forecasting in a competitive market: A neural network approach,”

Electric Power Systems Research, vol. 77, no. 10, pp. 1297–1304, 2007.

- [114] C. Garcia-Martos, J. Rodriguez, and M. Sanchez, “Mixed models for short-run forecasting of electricity prices: Application for the spanish market,” *IEEE Trans. on Power Systems*, vol. 22, no. 2, pp. 544–552, 2007.
- [115] T. Esum and P. Chapman, “Comparison of photovoltaic array maximum power point tracking techniques,” *IEEE Trans. on Energy Conversion*, vol. 22, no. 2, pp. 439–449, 2007.
- [116] K. Kobayashi, I. Takano, and Y. Sawada, “A study on a two stage maximum power point tracking control of a photovoltaic system under partially shaded insolation

conditions,” *IEEE Power Engineering Society General Meeting*, vol. 4, 2003.

- [117] K. Irisawa, T. Saito, I. Takano, and Y. Sawada, “Maximum power point tracking control of photovoltaic generation system under non-uniform insolation by means of monitoring cells,” *Conference Record of the IEEE Photovoltaic Specialists Conference*, pp. 1707–1710, 2000.

- [118] M. Bodur and M. Ermis, “A transceiver-aware routing framework for on-chip nanophotonic integration,” in *Proceedings of IEEE International Workshop on Emerging Technologies in VLSI CAD and Embedded Computing*, vol. 2, 1994, pp. 758–761.

- [119] E. Lorenz, J. Hurka, D. Heinemann, and H. Beyer,

“Irradiance forecasting for the power prediction of grid-connected photovoltaic systems,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 2, no. 3, pp. 2–10, 2009.

[120] M. Chary, N. Sreenivasulu, K. N. Rao, and D. Saibabu,

“Energy saving through vfds for fan drives in tobacco threshing plants,” in *Proceedings of IEEE International Conference on Industrial Technology 2000*, vol. 2, 2000, pp. 606–608.

[121] M. Pedrasa, T. Spooner, and I. MacGill, “Coordinated scheduling of residential distributed energy resources to optimize smart home energy services,” *IEEE Trans. on Smart Grid*, vol. 1, no. 2, pp. 134–144, 2010.

[122] KYOCERA Solar, Inc. (2010, Jun.) Online resource

of solar energy panel. [Online]. Available: <http://www.kyocerasolar.com/assets/001/5124.pdf>

[123] Solar Systems USA, Inc. (2012, Jun.) Online resource of solar panel cost. [Online]. Available: <http://www.solarsystemsusa.net/solar-panels>

[124] Solar Panel Info. (2010, Jun.) Online resource of solar panel life span. [Online]. Available: <http://www.solarpanelinfo.com/solarpanels/solar-panel-cost.php>

[125] T. Givler and P. Lilienthal, "Using homer software, nrels micropower optimization module, to explore the role of gen-sets in small solar power systems case study: Sri lanka," *Technical Report*, vol. NREL/TP-710-36774, pp. 1–23, 2005.

- [126] Public Renewables Partnership. (2012, Jun.) Online resource of solar energy panel cost. [Online]. Available: <http://www.repartners.org/solar/pvcost.htm>
- [127] Z. W. and F. L., "Critical peak pricing tariff design for mass consumers in great britain," in *Proceedings of 2011 IEEE Power and Energy Society General Meeting*, 2011, pp. 1–6.
- [128] E. Sortomme, M. Hindi, S. MacPherson, and S. Venkata, "Coordinated charging of plug-in hybrid electric vehicles to minimize distribution system losses," *IEEE Transactions on Smart Grid*, vol. 2, no. 1, pp. 198–205, 2011.
- [129] T. Terlaky, A. Vannelli, and H. Zhang, "On routing in vlsi design and communication networks," *Discrete Appl.*

Math., vol. 156, no. 11, pp. 2178–2194, Jun. 2008.

- [130] D. Miller, “Device requirement for optical interconnects to silicon chips,” in *Proceedings of IEEE, Special Issue on Silicon Photonics*, 2009, pp. 1166–1185.
- [131] D. Ding and D. Pan, “Oil: a nano-photonics optical interconnect library for a new photonic networks-on-chip architecture,” in *Proceedings of the 11th international workshop on System level interconnect prediction*, ser. SLIP '09, 2009, pp. 11–18.

Appendix

Appendix A Copyright Permission for from Chapters 2-5

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

- 1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
- 2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
- 3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

- 1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
- 2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
- 3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.